Herausgeber

T. LÄNGLE
M. HEIZMANN

FORUM BILDVERARBEITUNG 2022
IMAGE PROCESSING FORUM 2022



T. Längle | M. Heizmann (Hrsg.)

FORUM BILDVERARBEITUNG 2022
IMAGE PROCESSING FORUM 2022

FORUM BILDVERARBEITUNG 2022
IMAGE PROCESSING FORUM 2022

Herausgegeben von
T. Längle und M. Heizmann



Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu

Print on Demand 2022 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 2510-7224
ISBN 978-3-7315-1237-0
DOI 10.5445/KSP/1000150865

Vorwort

Bildverarbeitung ist definitionsgemäß die Wissenschaft von der Verarbeitung von Bildern. Damit verknüpft das Fachgebiet die Sensorik von Kameras – bildgebender Sensorik – mit der Verarbeitung der aufgenommenen Sensordaten – den Bildern. Aus dieser Verknüpfung resultiert der besondere Reiz dieser Disziplin. Bildern begegnet der Mensch ständig, schon weil das Sehen die wichtigste Informationsquelle als Handlungsgrundlage für den Menschen bildet. Durch die Verwendung von Kameras eröffnen sich aber noch weitergehende Chancen, da die Bildgebung nicht auf die biologischen Beschränkungen des Auges begrenzt sind: Hier sind beispielsweise die multi-/hyperspektrale Bilderfassung, hohe Bildraten oder die Maßverkörperung durch das Pixelraster der Kamera zu nennen. Da Kameras ähnlich anderen Produkten der Elektronik von der hohen Effizienz der Elektronikfertigung profitieren, werden auch hochwertige Kameras tendenziell immer günstiger.

In zahlreichen Aufgabenstellungen hat der Mensch eine intuitive Vorstellung, wie eine bestimmte Information aus einem Bild gewonnen werden kann. Beispiele sind die Erkennung von Defekten auf Oberflächen oder die Orientierung im Raum, für welche der Mensch unterschiedliche Auswertemethoden ganz intuitiv und ohne Kenntnis von einer konkreten „Algorithmik“ kombiniert. Die Verarbeitung von Bildern auf technische Systeme zu übertragen, kann immer noch herausfordernd sein. Während manche Aufgabenstellungen als weitgehend gelöst gelten, gibt es immer noch Herausforderungen, die dazu führen, dass Forschungsaktivitäten zu neuen Lösungen und erschließbaren Anwendungsfeldern führen. Hier zeigt sich ein weiterer Reiz der Bildverarbeitung, da Lösungsansätze oft Bausteine aus zahlreichen unterschiedlichen Disziplinen – von der Bildgebung über die Systemtheorie, die Signalverarbeitung bis hin zur Informationsfusion und zu maschinellem Lernen – zielführend verknüpfen.

Ziel des „Forums Bildverarbeitung“ ist es, solche interessanten Aufgabenstellungen und passende Lösungsansätze einem breiten Publikum zugänglich zu machen und den fachlichen Austausch zu den

zahlreichen Facetten der Bildverarbeitung anzuregen. Die Veranstaltung findet in jedem zweiten Jahr seit 2010 statt und wird inzwischen gemeinsam vom Geschäftsfeld Inspektion und Optronische Systeme des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildverarbeitung IOSB und dem Institut für Industrielle Informationstechnik IIIT des Karlsruher Instituts für Technologie KIT organisiert. Auch in diesem Jahr haben erfreulich viele Autoren dem Aufruf zur Einreichung von Beiträgen geantwortet. Der Programmausschuss konnte aus den Einreichungen nach sorgfältiger Begutachtung 24 hochwertige Beiträge auswählen und den Themenfeldern

- Bildgewinnung,
- Qualitätssicherung,
- Sortierung,
- Bildverarbeitung,
- Fahrzeuge sowie
- Mess- und Automatisierungstechnik.

zuordnen. Zur überwiegenden Zahl der Beiträge wurden Aufsätze erstellt, die im vorliegenden Tagungsband enthalten sind.

Wir danken den Autoren für ihre sorgfältig erstellten Aufsätze, den Mitgliedern des Programmausschusses für die aktive Ansprache von Autoren und ihre wertvolle Expertise bei der Begutachtung der Einreichungen und allen, die durch ihre Anwesenheit zum Gelingen des Forums Bildverarbeitung beitragen. Für die Organisation der Veranstaltung und die technische Unterstützung bei der Erstellung des Tagungsbands bedanken wir uns bei Britta Ost, Felix Lehnerer, Jürgen Hock und Alexander Enderle.

November 2022

Thomas Längle
Michael Heizmann

Wissenschaftliche Leitung

Prof. Dr.-Ing. T. Längle	Fraunhofer IOSB Karlsruhe
Prof. Dr.-Ing. M. Heizmann	Karlsruher Institut für Technologie

Programmausschuss

Prof. Dr. C. Bach	Buchs
Dr.-Ing. S. Bauer	Boston (Massachusetts)
Prof. Dr.-Ing. J. Beyerer	Karlsruhe
Prof. Dr. A. Braun	Düsseldorf
Dr. rer. nat. J. Eggert	Offenbach
Dr. M. Glitzner	München
Dr. T. Haist	Stuttgart
Prof. Dr. A. Heinrich	Aalen
Prof. Dr. B. Jähne	Heidelberg
Dipl.-Ing. M. Maurer	Wiesbaden
Dr. M. Overdick	Waldkirch
Prof. Dr. F. Salazar	Madrid (Spanien)
Dipl.-Ing. M. Stelzl	Mainz
Prof. Dr.-Ing. C. Stiller	Karlsruhe
Prof. Dr.-Ing. R. Tutsch	Braunschweig
Prof. Dr.-Ing. M. Ulrich	Karlsruhe
Prof. Dr.-Ing. S. Werling	Mannheim
Prof. Dr.-Ing. V. Willert	Schweinfurt

Contents

Preface i

Image acquisition

Mehrwellenlängen-Verfahren zur strukturierten Beleuchtung 1
M. Petz, P.-F. Hagen und R. Tutsch

Physics enhanced neural network for phase imaging using two axially displaced diffraction patterns 15
R. Li, G. Pedrini, L. Cao, and S. Reichelt

Areal multispectral sensor with variable choice of spatial and spectral resolution 25
T. Haist, R. Hahn, and S. Reichelt

Blurred resolution enhancement by graphene nanoplates 37
L. Carrilero, J. R. Castro, S. Pérez, T. Belenguer, and F. Salazar

Quality assurance

Optimal human labelling for anomaly detection in industrial inspection 49
T. Zander, Z. Pan, P. Birnstill, and J. Beyerer

Quality control of laser welds based on the weld surface and the weld profile 61
J. Hartung, A. Jahn, and M. Heizmann

Semantic segmentation with small training datasets: A case study for corrosion detection on the surface of industrial objects .. 73
D. Haitz, P. Hübner, M. Ulrich, S. Landgraf, and B. Jutzi

Innovative Qualitätssicherung mittels optimierter Bildverarbeitungsketten auf Basis von Deep Learning 87
K. Anding, G. Polte, L. Steinert, D. Garten, M. Kraft, M. Welzenbach und C. Gärtner

Real-time multi-image vignetting and exposure correction for image stitching 101
C. Kinzig, G. Feng, M. Granero, and C. Stiller

Sorting

Machine learning-based multiobject tracking for sensor-based sorting 115
G. Maier, M. Reith-Braun, A. Bauer, R. Gruna, F. Pfaff, H. Kruggel-Emden, T. Längle, U. D. Hanebeck, and J. Beyerer

Fast and comprehensive FPGA based BLOB analysis with the Hybrid-BLOB concept 127
S. Wezstein, M. Stelzl, and M. Heizmann

Image processing

An introduction to quantum image processing on real superconducting quantum computers 139
A. Geng, A. Moghiseh, K. Schladitz, and C. Redenbach

High-performance image reconstruction algorithm in CUDA C++ for ultra wideband multi-channel MIMO radar systems 151
J. Perske, H. Cetinkaya, C. Schwäbig, and S. Gütgemann

Deep learning and active learning based semantic segmentation of 3D CT data 163
M. Michen and U. Haßler

Vehicles

Signal processing pipeline for an autonomous electrical race car .. 177
M. Scheffler, O. Kettern, O. Zbaranski, F. Schäfer, K. Schmidt, B. Eberhardt, and S. Werling

Indoor floorplan estimation from 3D point clouds for *Scan-to-BIM* 189
O.H. Ramírez-Agudelo, A. Alex, L. Schreiber, N. Niemann, E. Milana, and C. Hammer

Measurement and automation technology

Perspektiveninvariante Inferenz von Eckpunkten in Packmustern von Kartonagen 201
F. Endres, L. Reinhart, T. Kaupp und V. Willert

Orts- und zeitaufgelöste bildbasierte Bestimmung der Brechungsindexverteilung bei der additiven Fertigung optischer Komponenten 215
M. Rank und A. Heinrich

Mehrwellenlängen-Verfahren zur strukturierten Beleuchtung
Multi-wavelength approach to structured illumination

Marcus Petz[1], Paul-Felix Hagen[2] und Rainer Tutsch[1]

[1] Technische Universität Braunschweig, Institut für Produktionsmesstechnik, Schleinitzstraße 20, 38102 Braunschweig
[2] Institut für Mess- und Regelungstechnik, Leibniz Universität Hannover, An der Universität 1, 30823 Garbsen

Zusammenfassung Auf dem Prinzip der strukturierten Beleuchtung basierende optische Messverfahren wenden häufig Phasenschiebeverfahren zur optischen Ortskodierung an. Während diese Ansätze in vielen Anwendungen eine effiziente und hochpräzise Kodierung ermöglichen, stoßen sie insbesondere bei der Überlagerung verschiedener Signalanteile an ihre Grenzen. Derartige Signalüberlagerungen entstehen bei der Streifenprojektion etwa durch Mehrfachreflexionen an der Werkstückoberfläche oder bei deflektometrischen Verfahren durch die Überlagerung von Vorder- und Rückseitenreflexen an transparenten Prüflingen. Vor dem Hintergrund dieser Problematik wird im vorliegenden Beitrag ein neuartiger Ansatz auf dem Gebiet der strukturierten Beleuchtung vorgestellt, welcher – basierend auf vergleichbaren Ansätzen aus dem Bereich der absolutmessenden Interferometrie – eine räumliche Kodierung durch eine Musterfolge mit ansteigender Ortsfrequenz umsetzt. Neben den Grundlagen des Verfahrens werden erste experimentelle Ergebnisse vorgestellt, welche aufzeigen, dass das Verfahren eine hohe Genauigkeit ermöglicht und zudem die Trennung überlagerter Signale mit hoher Qualität gelingt.

Schlüsselwörter Strukturierte Beleuchtung, optische Ortskodierung, Mehrwellenlängen-Kodierung, Streifenprojektion, Deflektometrie

Abstract Optical measuring methods based on the principle of structured illumination frequently apply phase shift evaluation for optical spatial coding. While these approaches allow for efficient and high-precision coding in many applications, they reach their limits in particular when different signal components are superimposed. This kind of signal superimpositions occur in stripe projection, for example due to multiple reflections on the workpiece surface, or in deflectometric methods due to the superimposition of front and rear side reflections on transparent samples. Against the background of this problem, a new approach in the field of structured illumination is presented in this article, which – based on comparable approaches from the field of absolute measuring interferometry – implements a spatial coding by a pattern sequence with increasing spatial frequency. In addition to the basics of the method, first experimental results are presented, which show that the method enables a high level of accuracy and that the separation of superimposed signals succeeds with high quality.

Keywords Structured illumination, optical spatial coding, multi-wavelength coding, fringe projection, deflectometry

1 Einleitung

Bei optischen Messverfahren wie der auf photogrammetrischen Prinzipien basierenden Streifenprojektion oder der phasenmessenden Deflektometrie wird eine optische Ortskodierung benötigt. Im Fall der Streifenprojektion erfolgt diese durch Projektion geeigneter Muster auf die Werkstückoberfläche, im Fall der Deflektometrie wird hingegen meist ein als Referenzmusterebene dienender Monitor mit entsprechenden Mustern beaufschlagt.

Die überwiegende Zahl der dabei zur optischen Kodierung genutzten Verfahren basiert auf dem Phasenschiebeprinzip, bei welchem eine definierte Anzahl sinusförmiger Streifenmuster mit einheitlicher Ortsfrequenz aber mit um definierte Winkel verschobener Phasenlage als Mustersequenz aufgezeichnet wird [1]. Hieraus lässt sich zunächst eine 2π-periodische relative Phase als im Messbereich mehrdeutige Ortsinformation berechnen.

Zur Entfaltung der relativen Phase sind unterschiedliche Ansätze gebräuchlich, wobei die Wiederholung der Phasenschiebemessung mit in der Regel drei geringfügig unterschiedlichen Ortsfrequenzen und die Auswertung der daraus resultierenden Schwebungssignale als ein vorteilhafter Ansatz erscheint [2]. In Kombination mit dem auf diesem Anwendungsgebiet gebräuchlichsten Phasenschiebeansatz, dem symmetrischen 4-Schritt-Algorithmus, besteht eine vollständige Mustersequenz zur Kodierung entlang einer Koordinatenachse demnach aus 12 Bildern – drei unterschiedliche Ortsfrequenzen jeweils mit den Phasenlagen 0°, 90° 180° und 270°. Dieser Kodierungsansatz wird im Folgenden als Referenz herangezogen.

Die umrissenen Phasenschiebeverfahren zeichnen sich durch eine vergleichsweise kurze Messdauer aufgrund der überschaubaren Musteranzahl, durch eine wenig rechenintensive und somit schnelle algorithmische Messdatenauswertung und nicht zuletzt durch eine hohe Auflösung und Genauigkeit der Ortskodierung aus. Eine in der praktischen Anwendung nicht unproblematische Eigenschaft dieser Verfahrensklasse besteht jedoch darin, dass sie nicht robust gegenüber der Überlagerung unterschiedlicher Signalanteile ist [3]. Werden also unterschiedliche Ortsbereiche der Muster auf dieselbe Stelle der Oberfläche beziehungsweise des Detektors abgebildet, resultiert daraus eine nicht behebbare Verfälschung der Ortsinformation.

Eine entsprechende Signalüberlagerung tritt etwa bei der Streifenprojektion auf, wenn das projizierte Licht an der Werkstückoberfläche teilweise gerichtet reflektiert wird und in der Folge auf einen anderen Bereich der Werkstückoberfläche trifft [4]. Im Fall der Deflektometrie tritt eine Signalüberlagerung insbesondere dann auf, wenn transparente Objekte wie etwa optische Linsen in Reflexion gemessen werden sollen [3]. In der Regel wird im Bild der Kamera dann eine Überlagerung von Vorder- und Rückseitenreflex beobachtet. Um dieses Problem abzumildern sind sowohl für die Streifenprojektion als auch für die Deflektometrie Ansätze beschrieben, bei welchen das Muster derart lokal maskiert wird, dass eine Signalüberlagerung soweit wie möglich vermieden wird [3, 4]. Dieses Vorgehen erhöht jedoch in jedem Fall die Messdauer signifikant, da die Messung mit einer mehr oder weniger hohen Anzahl unterschiedlich maskierter Muster wiederholt werden muss. Zudem ist die Bestimmung einer optimalen, an den jeweiligen Prüfling angepassten Maskierungssequenz zeitaufwendig und nicht

trivial.

Im vorliegenden Beitrag wird aus den genannten Gründen erstmals ein anderer, neuartiger Ansatz zur optischen Ortskodierung mittels sinusförmiger Streifenmuster vorgestellt. Dieser ist inspiriert von Mehrwellenlängen-Verfahren wie sie auf dem Gebiet der absolutmessenden Interferometrie zur Anwendung kommen, bei welchen etwa mittels einer durchstimmbaren Laserquelle eine Abstandskodierung entlang der Strahlachse durchgeführt werden kann [5]. Das Grundprinzip besteht darin, dass beim Durchstimmen mit geeigneten Wellenlängen eine linear vom Abstand zur Strahlungsquelle abhängige Oszillationsfrequenz des Interferenzsignals detektierbar ist. Dieses Grundprinzip wird im Folgenden auf die einachsige Ortskodierung mittels einer in ihrer Ortsfrequenz variierten Mustersequenz übertragen. Es werden die Grundlagen des Kodierungsverfahrens sowie der algorithmischen Auswertung vorgestellt und es werden erste experimentelle Ergebnisse präsentiert, welche aufzeigen, dass das neuartige Kodierungsverfahren eine mit dem oben beschriebenen Heterodyn-Phasenschiebverfahren vergleichbare Auflösung und Genauigkeit ermöglicht und dass ferner die Trennung überlagerter Signale mit hoher Güte gelingt.

2 Grundlagen des Mehrwellenlängen-Ansatzes

In der Interferometrie ist die Phase eines periodischen Signals in Abhängigkeit von der Wellenlänge λ, der Weglänge L und dem Brechungsindex n entsprechend Gleichung 1 bestimmt [6].

$$\phi\left(L\right) = \frac{2\pi}{\lambda/2} \cdot n \cdot L \tag{1}$$

Bei einer konstanten Wellenlänge λ und konstantem Brechungsindex n ergibt sich somit nach einer Weglänge L jeweils eine charakteristische Phase ϕ. Abhängig von der Phase kann mit Gleichung 2 die Signalintensität I berechnet werden, die zudem vom Interferenzkontrast γ abhängt [6].

$$I = I_0 \cdot \left[1 + \gamma \cdot \cos \frac{2\pi}{\lambda/2} \cdot n \cdot L\right] \tag{2}$$

Wird nun die Wellenlänge λ variiert und dabei die Weglänge L konstant gehalten, so ergibt sich über den betrachteten Zeitraum eine Phasenänderung. Wird die Wellenlänge λ derart durchgestimmt, dass $1/\lambda$ eine lineare Änderung erfährt, so ist die resultierende Phasenänderung gemäß Gleichung 1 linear. Eine lineare Phasenänderung ist gleichbedeutend mit einer harmonischen Schwingung mit einer weglängenabhängigen Frequenz. Die Frequenz und Phasenlage der resultierenden harmonischen Schwingung können gemessen werden und sind bei geeigneter Wahl der Wellenlängen λ_i ein lineares Maß für die Weglänge L.

Wird dieses aus der Interferometrie stammende Verfahren auf räumliche Signale übertragen, so entfällt zunächst der Faktor $1/2$ im Nenner von Gleichung 1, da bei der direkten Detektion von Ortsfrequenzen anders als in der Interferometrie keine doppelte Weglänge in Form von Hin- und Rückweg berücksichtigt werden muss. Anstelle der Weglänge L wird im Weiteren die Position X entlang der Kodierungsrichtung, also die Ortskoordinate innerhalb des Musters betrachtet. Für die betrachteten räumliche Signale entfällt zudem der Brechungsindex n. Entsprechend vereinfacht sich Gleichung 1 für den hier betrachteten Fall zu Gleichung 3.

$$\phi\left(X\right) = \frac{2\pi}{\lambda} \cdot X \tag{3}$$

An die Stelle des Interferenzkontrasts γ in Gleichung 2 tritt die Modulation M. Damit folgt für die beobachtbare Intensität $I\left(X\right)$ nachfolgende Gleichung 4.

$$I\left(X\right) = I_0 \cdot \left[1 + M \cdot \cos\frac{2\pi}{\lambda} \cdot X\right] \tag{4}$$

Eine Veranschaulichung einer entsprechenden Wellenlängensequenz, anhand welcher die weitere Diskussion nachvollzogen werden kann, ist in Abbildung 1 dargestellt. Zur Visualisierung wurden als Grenzen des Wellenlängenspektrums $\lambda_{\text{min}} = 20\,\text{px}$, $\lambda_{\text{max}} = 36\,\text{px}$ und die Anzahl N der diskreten Wellenlängen zu $N = 32$ gewählt. Die zwischen λ_{min} und λ_{max} liegenden Wellenlängen λ_i sind wie oben gefordert derart abgestuft, dass sich die Ortsfrequenz $1/\lambda_i$ linear ändert. In Abbildung 1 ist die resultierende Musterabfolge im Intervall $X \in [0\,\text{px}; 800\,\text{px}]$ dargestellt. Zunächst ist festzustellen, dass für $X = 0$ alle Signale dieselbe

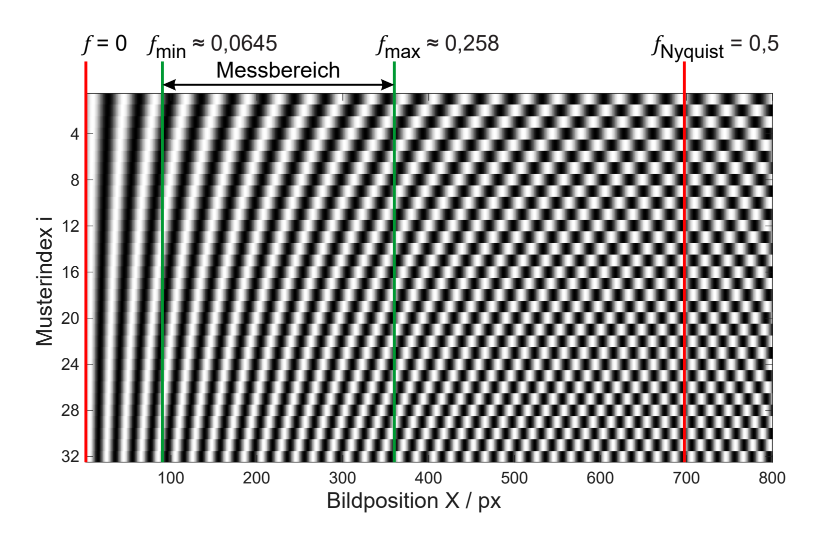


Abbildung 1: Beispielhafte Wellenlängen-Sequenz für Ortskodierung nach dem Mehrwellenlängen-Verfahren.

Anfangsphase aufweisen. Die am Ort $X = 0$ aus der Sequenz resultierende harmonische Schwingung weist daher die nicht auswertbare Frequenz $f = 0$ auf. Da sich die Frequenz mit zunehmender Koordinate X linear erhöht, wird irgendwann die korrekt erfassbare Nyquist-Frequenz f_{Nyquist} erreicht. Für die Frequenz f in Abhängigkeit der registrierten Periodenanzahl P der harmonischen Schwingung und der Anzahl N unterschiedlicher Musterwellenlängen gilt Gleichung 5.

$$f = \frac{P}{N-1} \tag{5}$$

Für das in Abbildung 1 exemplarisch betrachtete Muster ergibt sich für den Fall der Nyquist-Frequenz $f_{\text{Nyquist}} = 0,5$ die Anzahl der zwischen erstem und letztem Sample registrierten Perioden P durch Umstellen von Gleichung 5 somit zu $P = 0,5 \cdot 31 = 15,5$. Für die Phasendifferenz $\Delta\phi$ zwischen $\phi_{\lambda_{\text{min}}}$ und $\phi_{\lambda_{\text{max}}}$ gilt ausgehend von Gleichung 3 ferner allgemein nachfolgende Gleichung 6.

$$\Delta\phi = \phi_{\lambda_{\text{min}}} - \phi_{\lambda_{\text{max}}} = \left(\frac{2\pi}{\lambda_{\text{min}}} - \frac{2\pi}{\lambda_{\text{max}}} \right) \cdot X \tag{6}$$

Mit der für die Nyquistfrequenz hier geltenden Bedingung $\Delta\phi = 15,5 \cdot 2\pi$ kann der Ort X, an welchem die Nyquistfrequenz für obige Musterabfolge erreicht wird daher, wie in Abbildung 1 eingetragen, zu $X = 697,5$ px ermittelt werden. Jenseits dieser Koordinate fallen die Frequenzen wieder linear ab, während die Phasenlage um 180° gedreht ist. Innerhalb des theoretischen Frequenzbereichs von $f = 0$ bis $f = 0,5$ sind sinnvollerweise weitere Anforderungen an die aufgezeichneten Schwingungssignale zu stellen, welche den nutzbaren Ortsbereich weiter einschränken. So ist es im Interesse einer möglichst zuverlässigen Frequenz- und Phasenmessung etwa sinnvoll, eine Mindestanzahl aufgezeichneter Perioden P zu fordern. Mit der Forderung $P_{\text{min}} = 2$ ergibt sich für die betrachtete Sequenz eine Koordinate von $X_{\text{min}} = 90$ px. Um am anderen Ende des Messbereichs einen ausreichenden Abstand von der Nyquistfrequenz einzuhalten, ist zudem die Forderung einer Mindestanzahl an Samples pro Signalperiode zweckmäßig. Mit S als der Anzahl der Samples pro Signalperiode kann Gleichung 5 zu Gleichung 7 umgeschrieben werden.

$$f = \frac{P}{(N-1) \cdot S} \tag{7}$$

Mit der zweckmäßigen Forderung $S_{\text{min}} = 4$ ergibt sich somit $P_{\text{max}} = 8$ und in der Folge $X_{\text{max}} = 360$ px. Im vorliegenden Fall ergäbe sich demnach ein effektiv nutzbarer Messbereich vom $\Delta X = X_{\text{max}} - X_{\text{min}} = 270$ px. Die hier zur Veranschaulichung genutzten Parameter sind folglich für den praktischen Einsatz des Verfahrens nicht zweckmäßig gewählt. Mit den für die im Weiteren vorgestellten experimentellen Untersuchungen gewählten Parametern $\lambda_{\text{min}} = 20$ px, $\lambda_{\text{max}} = 21$ px, $N = 48$, $P_{\text{min}} = 2$ und $S_{\text{min}} = 4$ entsteht hingegen ein nutzbarer Kodierungsbereich von $\Delta X = X_{\text{max}} - X_{\text{min}} = 5040\,\text{px} - 840\,\text{px} = 4200$ px. Hiermit kann folglich selbst ein 4K Monitor eindeutig ortskodiert werden oder alternativ können, wie in Abschnitt 4.1 gezeigt, zwei Full HD Monitore mit nicht-überlappenden Frequenzbereichen kodiert werden.

3 Datenauswertung

Die rechnerische Auswertung der aufgenommen Bildsequenzen besteht im Wesentlichen aus der Bestimmung von Frequenz und Phase

der für jeden Bildpunkt aufgezeichneten harmonischen Schwingung. Im Folgenden werden zwei bereits experimentell näher untersuchte Fälle unterschieden, nämlich erstens die Messung ohne Überlagerung verschiedener Ortsinformationen, für welche die Bestimmung der Parameter nur einer harmonischen Schwingung erforderlich ist, sowie zweitens der Fall der Überlagerung zweier Ortsinformationen, für welchen die Parameter zweier unterschiedlicher harmonischer Schwingungen ermittelt werden müssen.

3.1 Auswertung einer Frequenz

Für jeden Messpunkt werden aus der Aufzeichnung der Bildsequenz N diskrete Intensitätswerte y_i gewonnen, welche in ihrer zeitlichen Abfolge eine harmonische Schwingung repräsentieren. Als Maß für die interessierende Ortskoordinate X dient, aufgrund des gegenüber der Frequenz besseren Signal-Rausch-Verhältnisses, der Phasenwinkel des Signals, wobei jedoch auch die Frequenz benötigt wird, um die Entfaltung der periodischen relativen Phase vornehmen zu können. Aufgrund der eher geringen Anzahl an Stützstellen N zeigt sich, dass eine zur Lösung des Problems naheliegende Fouriertransformation eine nur sehr geringe und letztlich trotz Interpolation nicht ausreichende Frequenzauflösung bietet. Es ist daher erforderlich, eine Sinusfunktion iterativ an die Messdaten anzupassen. Hierfür werden jedoch, um ein gutes Konvergenzverhalten zu erzielen, hinreichend gute Startwerte für die freien Parameter der Zielfunktion benötigt. Diese lassen sich mittels einer Fouriertransformation mit ausreichender Güte bestimmen, so dass der Auswerteprozess im Wesentlichen aus der Abfolge einer Fouriertransformation und eines iterativen Sinusfits besteht.

Offenkundig ist damit der Rechenaufwand für das vorgestellte Mehrwellenlängen-Verfahren signifikant höher als jener für das etablierte Phasenschiebeverfahren. In einer ersten Implementierung wurde das oben umrissene Auswerteverfahren in den beiden wesentlichen Teilen, der Fast Fourier Transformation (FFT) sowie des Sinusfits, in einer C++ Dynamic Link Library realisiert, wobei von der Möglichkeit der Parallelisierung von Teilaufgaben gebraucht gemacht wurde. Die bislang erzielten Auswertedauern liegen exemplarisch auf einem Prozessor vom Typ AMD Ryzen™ 7 5800H bei rund 5 Sekunden pro eine Million Messpunkte. Damit liegt die Auswertedauer bereits in

dieser frühen Erprobungsphase in einer durchaus praxistauglichen Größenordnung. Eine nochmals deutliche Reduzierung der Auswertedauer wäre mit überschaubarem Aufwand durch Nutzung von GPU Computing insbesondere für die FFT erreichbar.

3.2 Auswertung zweier Frequenzen

Sofern an einem Ort der Bildsequenz zwei Ortsinformation zur Überlagerung kommen, sind statt der Parameter für nur eine harmonische Schwingung die Parameter zweier harmonischer Schwingungen zu berechnen. Unter günstigen Randbedingungen – das heißt insbesondere sofern die überlagerten Frequenzen hinreichend weit voneinander entfernt liegen und eine ähnlich hohe Modulation aufweisen – ist es möglich, den zuvor beschriebenen Auswerteablauf im Grundsatz beizubehalten. In diesem Fall werden aus dem FFT-Spektrum die beiden lokalen Maxima mit den größten Amplituden extrahiert und direkt als Startwerte für das Optimierungsproblem genutzt. Es werden im Rahmen der Optimierung nach Gauß-Newton für jede der beiden Schwingungen eine individuelle Modulation, Frequenz und Phase angesetzt, während der Offset nur summarisch für beide Schwingungen berechenbar ist.

Die Erfahrungen verschiedener Testmessungen zeigen, dass für anspruchsvollere Szenarien – also insbesondere geringer Frequenzabstand und/oder deutlich unterschiedliche Modulation beider Signale – die Wahrscheinlichkeit für ein Scheitern dieses direkten Ansatzes deutlich zunimmt. Zum einen werden dann mit zunehmender Häufigkeit zu stark abweichende Startwerte aus der FFT ermittelt, zum anderen zeigt das Gauß-Newton-Verfahren zunehmend problematisches Konvergenzverhalten. Das Zusammenwirken von FFT und Sinusfit kann jedoch im Grundsatz beibehalten werden, nur dass dieses vorteilhafterweise in mehrere Teilschritte untergliedert wird und als finaler Optimierungsschritt ein Downhill-Simplex-Verfahren eingesetzt wird.

4 Messergebnisse

Die bislang nach dem vorgestellten Ansatz durchgeführten Messungen verfolgen im Wesentlichen zwei Ziele. Erstens soll untersucht werden,

ob das Verfahren grundsätzlich eine vergleichbar hohe Kodierungsgüte wie die etablierten Phasenschiebeverfahren ermöglicht. Zweitens soll überprüft werden, ob eine Trennung zunächst zweier überlagerter Ortsinformationen – wie sie typischerweise bei der deflektometrischen Linsenmessung auftritt – mit grundsätzlich vergleichbarer Qualität wie bei einer Messung ohne Überlagerung möglich ist.

4.1 Messung ohne Signalüberlagerung

Der Messaufbau des Szenarios ohne Signalüberlagerung besteht aus einer elektronischen Kamera vom Typ IDS UI5240SE-M mit einem Objektiv FUJINON HF9HA-1B, welche direkt die vollständige Bildschirmfläche eines Samsung PLS-Monitors vom Typ S24E650 mit einer Auflösung von 1920 *x* 1200 px beobachtet. Als Referenzkodierungsansatz wird die heterodyne Phasenschiebetechnik nach [2] mit Wellenlängen von $\lambda_1 = 20\,\text{px}$, $\lambda_2 = 21,5\,\text{px}$ und $\lambda_3 = 23\,\text{px}$ verwendet. Für den Mehrwellenlängen-Ansatz wurde der Parametersatz $\lambda_{\text{min}} = 20\,\text{px}$, $\lambda_{\text{max}} = 21\,\text{px}$, $N = 48$ und $P_{\text{min}} = 5$ gewählt. Die vollständige Phasenschiebungssequenz besteht somit aus insgesamt 12 Bildern, während die Mehrwellenlängensequenz aus 48 Bildern besteht. In beiden Fällen wird jedes Bild durch Addition von vier 12-Bit-Bildern der Kamera erhalten, wodurch Sätze synthetischer 14-Bit-Bilder erzeugt werden.

Als Maß für die Genauigkeit wird der erhaltene relative Positionsfehler verwendet. Dieser sei hier definiert als das Verhältnis von Phasenabweichungen $\Delta\Phi$ und der Spanne der Phasenwerten $\Phi_{\text{max}} - \Phi_{\text{min}}$ in jeder Messung. Da die realen Messungen nicht nur hochfrequente, rauschartige Abweichungen enthalten, sondern auch niederfrequente Abweichungen, die sich aus der Aufbaugeometrie und optischen Verzerrungen ergeben, wird der relevante hochfrequente Anteil durch Hochpassfilterung der Phasendaten bestimmt. Beide Messungen liefern ca. 654.000 Messpunkte und einen in sehr guter Näherung normalverteilten relativen Positionsfehler. Dabei beträgt die Standardabweichung für den Mehrwellenlängen-Ansatz ca. $7,79 \cdot 10^{-6}$, während das Phasenschiebeverfahren einen Wert von ca. $8,54 \cdot 10^{-6}$ liefert. Allerdings sollte hierbei beachtet werden, dass die Anzahl der Bilder beim Mehrwellenlängen-Ansatz um den Faktor 4 größer ist. Dennoch lässt sich festhalten, dass das Mehrwellenlängen-Verfahren trotz des etwas höheren Messaufwand eine gegenüber etablierten Phasenschiebever-

fahren konkurrenzfähige Ortskodierung ermöglicht.

4.2 Messung mit Signalüberlagerung

Für die Untersuchungen zur Trennbarkeit zweier überlagerter Signalanteile wurde ein Aufbau aus zwei unter einem Winkel von 90° zueinander angeordneten Monitoren vom oben genannten Typ in Verbindung mit einer Kamera vom Typ IDS UI3070CP-M und einem Objektiv FUJINON HF16HA-1B eingesetzt. Unmittelbar vor dem Objektiv ist ein Strahlteilerwürfel derart positioniert, dass die Kamera eine Überlagerung der beiden Displays beobachtet, sobald beide aktiviert sind. Anhand dieses Aufbaus wurden zwei Szenarien untersucht. Zum ersten der Fall, dass die beiden Monitore mit einer Mustersequenz beaufschlagt werden, welche zu einem nicht-überlappenden Frequenzbereich beider Monitore führt. Zum zweiten das anspruchsvollere Szenario, dass beide Monitore mittels desselben Frequenzbandes kodiert werden.

Der für das erste Szenario verwendete Parametersatz lautet wie zuvor $\lambda_{\text{min}} = 20\,\text{px}$, $\lambda_{\text{max}} = 21\,\text{px}$, $N = 48$. Mit $P_{\text{min},1} = 2,4$ für den in Transmission durch den Strahlteiler beobachteten Monitor 1 und $P_{\text{min},2} = 7,4$ für den gespiegelt beobachteten Monitor 2 liefert das Muster zwei nicht überlappende Frequenzbereiche, wovon der erste bei $X_{\text{min},1} = 1176\,\text{px}$ und der zweite bei $X_{\text{min},2} = 3108\,\text{px}$ beginnt.

Um die Güte der Signaltrennung zu bewerten, wurden zusätzlich Messungen mit jeweils nur einem aktivierten Monitor durchgeführt, so dass Referenzdaten ohne Signalüberlagerung zur Verfügung stehen. Die Differenzen zwischen den rechnerisch separierten Phasendaten aus der Messung mit Überlagerung sowie den jeweils korrespondierenden Einzelmessungen zeigen eine hervorragende Übereinstimmung der Phaseninformation. Der relative Phasenfehler, zu verstehen als $\Delta\Phi_{\text{rel}} = \Delta\Phi/2\pi$, weist für Monitor 1 eine Standardabweichung von $\sigma_{\Delta\Phi_{\text{rel}},1} \approx 1/1107$ und für den deutlich dunkler erscheinenden Monitor 2 von $\sigma_{\Delta\Phi_{\text{rel}},2} \approx 1/460$ auf. Diese Werte stimmen ungefähr mit jenen überein, die ausgehend vom Grundrauschen des Verfahrens auch bei Subtraktion zweier Einzelmessungen ohne Signalüberlagerung zu erwarten wären.

Werden beide Monitore mit demselben Frequenzbereich kodiert, so gelingt die Trennung der überlagerten Signalanteile im größten Teil

des Bildfeldes vergleichbar gut, wie zuvor beschrieben, da wegen des Strahlteilers Monitor 2 gegenüber Monitor 1 horizontal gespiegelt erscheint. Lediglich in einem schmalen Bereich, in welchem die Frequenzen beider Signalanteile sehr dicht beisammen liegen, ermöglicht das derzeitige Berechnungsverfahren keine erfolgreich Signaltrennung. Der im beschriebenen Messaufbau für eine erfolgreiche Signaltrennung derzeit erforderliche Signalversatz entspricht etwa 100 Pixel in der Monitorebene. Es ist davon auszugehen, dass eine Optimierung des Berechnungsverfahrens eine weitere Steigerung der Trennschärfe ermöglicht.

5 Zusammenfassung

Das vorgestellte Mehrwellenlängen-Verfahren stellt einen neuartigen Ansatz zur strukturierten Beleuchtung dar, welcher bei Messverfahren wie der Streifenprojektion und der Deflektometrie eingesetzt werden kann. Das Hauptmerkmal des Ansatzes besteht darin, dass er im Gegensatz zu den etablierten Phasenschiebetechniken mit der Überlagerung mehrerer Ortsinformationen umgehen kann. Neben den Grundlagen des Ansatzes werden Datenauswertungsverfahren für einzelne und doppelte Ortsinformationen pro Bildpunkt aufgezeigt. Für beide Fälle liegen experimentelle Daten vor, die das Potenzial des Ansatzes aufzeigen. Es lässt sich festhalten, dass der Mehrwellenlängen-Ansatz eine optische Ortskodierung mit Unsicherheiten ermöglicht, welche vergleichbar mit jener etablierter Phasenschiebetechniken sind. Für den Fall der Signalüberlagerung zeigt das gegebene Beispiel, dass zwei überlagerte Datensätze effektiv und korrekt getrennt werden können, sofern die zu trennenden Ortsinformationen nicht zu ähnlich sind. Somit zeigt der Mehrwellenlängenansatz ein hohes Potenzial für spezielle Anwendungen im Bereich der Streifenprojektion und Deflektometrie, die mit den etablierten Phasenschiebetechniken nicht bewältigt werden können.

Literatur

1. C. Zuo, S. Feng, L. Huang, T. Tao, W. Yin, and Q. Chen, "Phase shifting algorithms for fringe projection profilometry: A review," *Optics and Lasers in*

Engineering, vol. 109, pp. 23–59, 2018.

2. M. Petz, H. Dierke, and R. Tutsch, "Wellenlängenoptimierung bei Heterodyn-Phasenschiebeverfahren," *tm - Technisches Messen*, vol. 87, no. 10, pp. 599–613, 2020.
3. C. Faber, *New Methods and Advances in Deflectometry*, ser. Progress in Modern Optics, S. Malzer, G. Leuchs, P. S. J. Russell, and V. Sandoghdar, Eds. Erlangen Scientific Press, Univ. Erlangen-Nürnberg, 2012.
4. P. Kayser, "Verfahren und Vorrichtung zum dreidimensionalen optischen Vermessen von Objekten mit einem topometrischen Messverfahren sowie Computerprogramm hierzu," Deutsches Patentamt München, 2013.
5. H. Kikuta, K. Iwata, and R. Nagata, "Distance measurement by the wavelength shift of laser diode light," *Appl. Opt.*, vol. 25, no. 17, pp. 2976–2980, Sep 1986.
6. F. Pollinger, K. Meiners-Hagen, and A. Abou-Zeid, "Absolutlängen mittels Mehrwellenlängen-Diodenlaserinterferometrie," *PTB-Mitteilungen*, vol. 120, no. 2, pp. 105–109, 2010.

Physics enhanced neural network for phase imaging using two axially displaced diffraction patterns

Rujia Li[1,2], Giancarlo Pedrini[1], Liangcai Cao[2], and Stephan Reichelt[1]

[1] Universität Stuttgart, Institut für Technische Optik, Pfaffenwaldring 9, 70569 Stuttgart, Germany
[2] Tsinghua University, Department of Precision Instruments, Qinghuayuan 1, 100084 Beijing, China

Abstract In this work, we propose a physics-enhanced two-to-one Y-neural network (two inputs and one output) for phase retrieval of complex wavefronts from two diffraction patterns. The learnable parameters of the Y-net are optimized by minimizing a hybrid loss function, which evaluates the root-mean-square error and normalized Pearson correlated coefficient on the two diffraction planes. An angular spectrum method network is designed for self-supervised training on the Y-net. Amplitudes and phases of wavefronts diffracted by a USAF-1951 resolution target, a phase grating of 200 lp/mm, and a skeletal muscle cell were retrieved using a Y-net with 100 learning iterations. Fast reconstructions could be realized without constraints or a priori knowledge of the samples.

Keywords Coherent diffraction imaging, phase retrieval, deep neural network

1 Introduction

Retrieving the phase from diffraction patterns is a long-standing problem. In the recorded intensity patterns, the object wavefront is superimposed with its a phase-conjugated and for reconstructing the wavefront without conjugation, the phase needs to be retrieved. Conventional methods used constraints to iteratively solve the phase retrieval

problem. A priori knowledge of the object plane [1] or modulations applied on the imaging path [2] [3] can be the constraints. Optimization iterations are needed.

Deep learning is a powerful approach for solving optimization problems. A convolutional neural network (CNN) is trained with a dataset for mapping input to output. CNNs are widely used in image processing, they have an end-to-end structure, which can be trained to retrieve a phase pattern from an intensity pattern [4] [5]. After training on a dataset, the reconstruction can be directly made by a CNN without further optimization. The phase retrieval problem has an explicit physical model and a CNN can be enhanced with the diffraction principle [5] in order to avoid training with thousands of patterns. However, the end-to-end structure of a CNN described in [6] limits the object to be phase-only. Splicing the phase and amplitude into one image seems to be a straightforward solution, but a CNN uses a convolution kernel for feature extraction. The connected edges of the amplitude and phase pattern may be convoluted with one kernel and generate data against the physical model.

In this work, we propose a physics-enhanced neural network for retrieving a complex wavefront from two axially displaced diffraction patterns. A two-to-one Y-net (two inputs and one output) is designed to retrieve the phase on the first plane. Then the complex wavefront is calculated with the retrieved phase and the square root of the recorded intensity pattern. An angular spectrum method (ASM) network is designed to calculate the wave propagation. The Y-net is trained with the diffraction between the two recording planes and produces a phase on the first plane, which can be used to generate two patterns on the two recording planes. The errors between generated and recorded patterns are evaluated with a hybrid loss function. The normalized Pearson correlation coefficient and root mean square error are used to build the hybrid loss function. The learnable parameters in the Y-net are optimized by gradient descent on the hybrid loss function. After training on a dataset, the Y-net can be generalized to retrieve complex wavefronts without optimization. Reconstruction can also be made using an untrained Y-net. An amplitude-only UASF-1951 resolution chart, a phase grating, and a skeletal muscle cell are experimentally reconstructed using an untrained Y-net.

2 Y-net for retrieving the complex wavefront

A schematic of the setup used for recording axially displaced diffraction patterns is shown in Fig 1 . The sample is illuminated by a plane wave and the diffraction patterns are recorded on two planes at distances of z′and z′+z. To reconstruct the complex-valued object, the phase on the two diffraction patterns is retrieved using a Y-net.

The proposed Y-net is a fusion of two U-nets. There are two down-sampling paths and one up-sampling path, which are composed of four down-sampling and corresponding up-sampling convolution blocks. In each convolution block, the information passes downstream along with two sets of batch normalization layers, rectified linear unit (ReLU) layer, and a convolution layer. The feature maps in each down-sampling block are extracted using a 3×3 convolution kernel with a stride of 2. In the bottleneck of the Y-net, the feature maps from the two down-sampling paths are connected as the input of the up-sampling path. Then the up sampling is made with transposed convolutions. There are residual layers and skip connections after the convolution blocks to make the deep Y-net easy to optimize by avoiding the vanishing gradients problem and mitigating the degradation problem.

The schematic for training the Y-net is shown in Fig.1 (b). In the first training loop, the learnable parameters are randomly initialized. This initialization helps keeping the signal from expanding to an extremely high value or vanishing to zero. Then the learnable parameters are optimized by minimizing a hybrid loss function, which is built by following the optical diffraction model.

The hybrid loss function for the Y-net is a linear combination of the loss function on two diffraction patterns. The output of the Y-net is set to be the phase on the first diffraction pattern. The complex wavefronts on the two planes follow the Rayleigh-Sommerfeld diffraction. By merging the phase $\varphi(x_1, y_1)$ with the first recorded intensity $I_1(x_1, y_1)$, we obtain the wavefront on the first plane $u_1(x_1, y_1) = \sqrt{I_1(x_1, y_1)} exp[i\varphi(x_1, y_1)]$. After propagating $u_1(x_1, y_1)$ to the second plane, we obtain the wavefront $u_2(x_2, y_2) = prop_z\{u_1(x_1, y_1)\}$, where z is the distance between the two planes. For evaluating the differences between $u_2(x_2, y_2)^2$ and $I_2(x_2, y_2)$, a loss function is built from the linear combination of the root-mean-square error (RMSE) and the

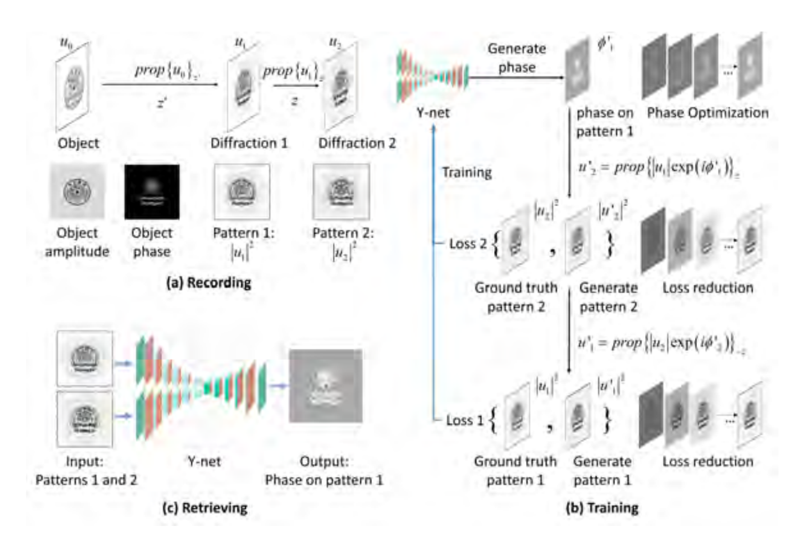


Figure 1: (a) Recording two patterns diffracted by a complex object; (b) Training the Y-net based on diffraction between the two planes; (c) Retrieving the phase on the first pattern.

normalized Pearson correlation coefficient (PCC),

$$Loss\{I, I'\} = l_{PCC} PCC\{I, I'\} + l_{RMSE} RMSE\{I, I'\} \tag{1}$$

$$PCC\{I, I'\} = \frac{1}{2}\{1 - \frac{\sum_{m,n}[I(m,n) - I_{ave}][I'(m,n) - I'_{ave}]}{\sqrt{\sum_{m,n}[I(m,n) - I_{ave}]^2 \sum_{m,n}[I'(m,n) - I'_{ave}]^2}}\} \tag{2}$$

$$RMSE\{I, I'\} = \sqrt{\frac{\sum_{m,n}[I(m,n) - I'(m,n)]^2}{MN}} \tag{3}$$

where l_{PCC} and l_{RMSE} are the relative weights of the normalized PCC and RMSE, m and n are integer numbers, M and N are the numbers of pixels in the patterns, I_{ave} and I'_{ave} are the average pixel values of the images. The PCC measures the linear similarity between the two patterns, which is evaluated by the ratio between the covariance of the pixel values and the product of their standard deviations. The PCC has a value between -1 and 1, where 1 represents two similar patterns. To perform gradient descent, the PCC operator is normalized as shown

in Eq.2. The normalized PCC has a value between 0 and 1, where 0 represents a high similarity. The RMSE is used together with PCC to obtain a better convergence in a training loop. The RMSE compares every pixel value on the generated intensity and the captured ground truth. The scaling effect of the PCC can be reduced by using the RMSE evaluation. When the RMSE value is 0, the generated intensity and the captured ground truth are the same on every pixel of the image.

In order to apply a sufficient constraint to the neural network, the loss function is also built on the second plane. The amplitude of the propagated wave $u_2(x_2, y_2)$ is replaced by $\sqrt{I_2(x_2, y_2)}$. Then the updated wavefront $u_2'(x_2, y_2)$ is propagated to the first plane, $u_1'(x_1, y_1) = prop_{-z}\{u_2'(x_2, y_2)\}$. The differences between $u_1'(x_1, y_1)^2$ and $I_1(x_1, y_1)$ are evaluated. The hybrid loss function for training the Y-net is $d_1 Loss_1\{I_1\} + d_2 Loss_2\{I_2\}$, where d_1 and d_2 are the weights of the loss on the two diffraction planes. Training the neural network is a process of optimizing the weights of each layer to minimize the prediction error between the outputs and ground truth. This is usually made by using gradient descent methods on the loss functions. In this work, the ADAM optimization is used for minimizing the hybrid loss function on the two planes. A well-trained Y-net retrieves a phase following the diffraction principles between the two planes.

3 Reconstructions in experiments

Experimental results were obtained by using an amplitude-only USAF-1951 resolution test target, a phase grating, and a skeletal cell sample. The diffraction patterns were recorded using the setup shown in Fig. 2(a). The samples were illuminated with a plane wave having wavelength 655 nm. The pixel size of the camera is 2 μm. After capturing the first diffraction pattern, the camera was shifted for capturing the second. The distance between the two planes was 400 μm. The size of the diffraction patterns was 512×512 pixel, this is a compromise for obtaining good resolution under fast training. Better results could be obtained using more pixels, but in this case a longer training time would be necessary.

Figs. 2(b) and (c) show the recorded patterns of the USAF-1951 resolution target. The first pattern was recorded at 4.4 mm distance

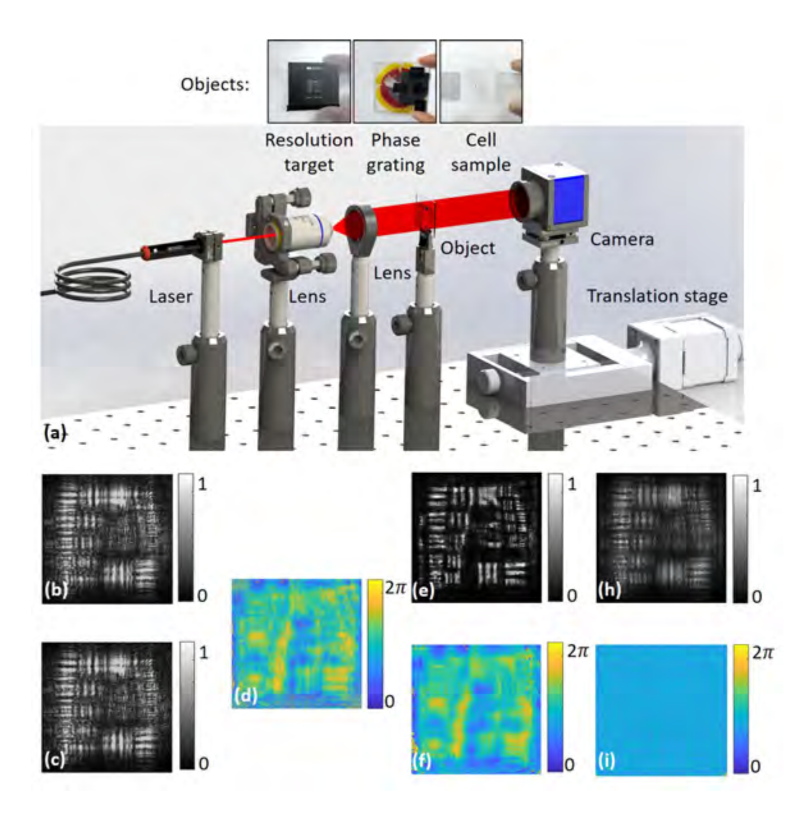


Figure 2: Experimental reconstruction for an amplitude-only USAF-1951 resolution target. (a) Schematic of the experimental setup; (b) and (c) The recorded diffraction patterns on the two planes; (d) Retrieved phase on the plane of (b); (e) and (f) Amplitude and phase of the reconstruction using the Y-net with 100 iterations; (h) and (i) Amplitude and phase of the reconstruction by propagating the first diffraction pattern.

from the object. An untrained Y-net was used for reconstruction. Self-supervised learning was performed by optimizing the hybrid loss function with 100 iterations. With the ASM network, the Y-net learns to retrieve a phase following the diffraction between the two recording planes. As shown in Fig. 2(d), the feature of the object can be distinguished from the retrieved phase. The complex wavefront on the first plane is calculated by multiplying the retrieved phase and the recorded amplitude. The phase and amplitude components are then reconstructed after propagating the calculated wavefront to the object plane. The intensity and the phase of the reconstruction are shown in Figs. 2(e) and (f). The sixth element of group five in the USAF-1951 target was resolved (line width of 8.77 μm). The reconstruction of the complex wavefront was made using the untrained Y-net without a priori knowledge. Figs. 2(h) and (i) shows the reconstruction of intensity and phase obtained by simply propagating the first diffraction pattern to the object plane. The intensity is not correctly reconstructed due to the presence of the conjugated wavefront.

A phase grating was also investigated with the same experimental setup shown in Fig. 2(a). The phase grating has a period of 5 μm (200 lp/mm). The first pattern was captured at a distance of 4.6 mm from the phase grating. Then the camera was shifted 400 μm for recording the second pattern. After self-supervised learning (100 iterations), the phase distributions of the gratings was reconstructed (see Figs. 3(b)). In this experiment, the phase grating cannot be reconstructed using simple propagation of the recorded diffraction pattern (Figs. 3(c), (d)).

A skeletal muscle cell was used in another experiment, to further demonstrate the capability of the Y-net. In this case the sample was illuminated with a plane wave having wavelength of 632.8 nm. The pixel size of the camera was 5.86 μm. The first diffraction pattern was captured 39.4 mm away from the specimen, this distance was numerically determined by back propagating the retrieved wave from the recording plane to the object plane. Then the camera was shifted 1 mm for capturing the second pattern. The phase at the first plane was retrieved after training the Y-net with 100 iterations. The reconstruction of the sample is obtained by propagating the retrieved wavefront. The reconstructed amplitude and phase of the skeletal muscle cell are shown in Figs. 3(e) and (f). The amplitude and phase show different structures of the skeletal muscle.

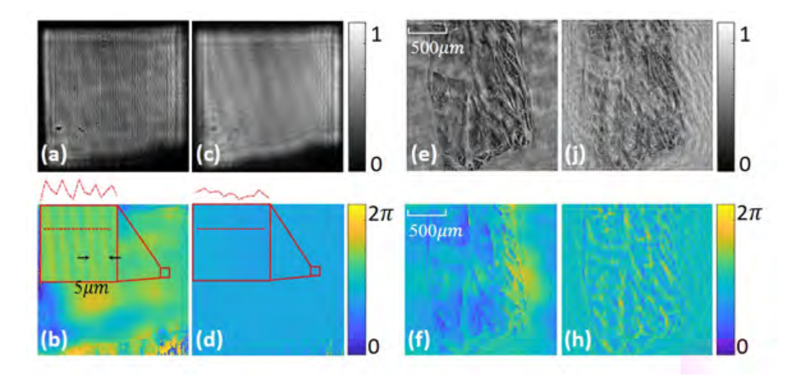


Figure 3: Experimental reconstruction for the phase grating and skeletal muscle cell sample. (a), (e) and (b), (f) Amplitude and phase of the reconstruction using the Y-net with 100 iterations; (c), (j) and (d), (h) Amplitude and phase of the reconstruction by propagating the first diffraction pattern.

4 Conclusion

Y-net is proposed to efficiently reconstruct complex wavefronts. With self-supervised training through an ASM network, the Y-net learns the diffraction between the two planes. Only two diffraction patterns are needed for the reconstruction. The two patterns may also be simultaneously captured using two cameras and one beam splitter. Then a well-trained Y-net may realize a quasi-real-time phase retrieval. The Y-net can be trained on a big dataset for the best generalization. The Y-net has a promising potential in the investigation of both timely and spatially varying physical processes. The large-scale complex wavefront can be rapidly retrieved using a well-trained Y-net. Besides the optical diffraction, this two-to-one Y-net may also be applied on learning other physical principles, such as the transmission of sound wave.

References

1. J. R. Fienup, "Phase retrieval algorithms: a comparison," *Appl. Opt.*, vol. 21, no. 15, pp. 2758–2769, Aug 1982. [Online]. Available: http://opg.optica.org/ao/abstract.cfm?URI=ao-21-15-2758

2. F. Zhang, G. Pedrini, and W. Osten, “Phase retrieval of arbitrary complex-valued fields through aperture-plane modulation,” *Phys. Rev. A*, vol. 75, p. 043805, Apr 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.75.043805
3. R. Li and L. Cao, “Complex wavefront sensing based on alternative structured phase modulation,” *Appl. Opt.*, vol. 60, no. 4, pp. A48–A53, Feb 2021. [Online]. Available: https://opg.optica.org/ao/abstract.cfm?URI=ao-60-4-A48
4. A. Sinha, J. Lee, S. Li, and G. Barbastathis, “Lensless computational imaging through deep learning,” *Optica*, vol. 4, no. 9, pp. 1117–1125, Sep 2017. [Online]. Available: https://opg.optica.org/optica/abstract.cfm?URI=optica-4-9-1117
5. F. Wang, C. Wang, C. Deng, S. Han, and G. Situ, “Single-pixel imaging using physics enhanced deep learning,” *Photon. Res.*, vol. 10, no. 1, pp. 104–110, Jan 2022. [Online]. Available: https://opg.optica.org/prj/abstract.cfm?URI=prj-10-1-104
6. F. Wang, Y. Bian, H. Wang, M. Lyu, G. Pedrini, W. Osten, G. Barbastathis, and G. Situ, “Phase imaging with an untrained neural network,” *Light: Science&Applications*, vol. 9, no. 77, May 2020. [Online]. Available: https://doi.org/10.1038/s41377-020-0302-3

Areal multispectral sensor with variable choice of spatial and spectral resolution

Tobias Haist, Robin Hahn, and Stephan Reichelt

Universität Stuttgart, Institut für Technische Optik,
Pfaffenwaldring 9, 70569 Stuttgart, Germany

Abstract We present a newly developed method for snapshot multispectral imaging. The core idea is to use a diffractive optical element (DOE) in an intermediate image plane. The main advantages are the potentially cost effective implementation for different applications, e.g. for classification and the possibility to use different spatio-spectral samplings at different field positions. By appropriate choice of the DOE it is possible to chose the spectral and spatial sampling pattern. We also shortly address the issue of light efficiency for different approaches towards multispectral imaging.

Keywords Hyperspectral imaging, multispectral imaging, diffractive optics

1 Introduction

Spatially resolved spectral information can be fruitfully employed in a lot of applications ranging from food monitoring to the detection of air pollution. Most often, image sensors with so-called Bayer patterns are used which mimic the human visual system with three broad spectral channels, typically denoted as the “short” (blue), “middle” (green) and “long” (red) bands.

For some applications other or more spectral bands are advantageous. But one has to keep in mind that there is always a trade-off between spectral resolution, number of spectral channels, spatial resolution, light efficiency and measurement time. If high spectral and spatial resolution is desired, typically, the amount of light per spatio-spectral pixel element is low (for a given entrance pupil and luminance

of the scene to be imaged). We can discriminate between "hyper-" and "multi-"spectral imaging based on the number of spectral channels. Typically, "hyper" is used for a lot of channels (e.g. more than 10). In the following we use the term "multi" if more than one channel is used. Therefore, even a standard RGB sensor is defined to be a "multi-spectral" sensor.

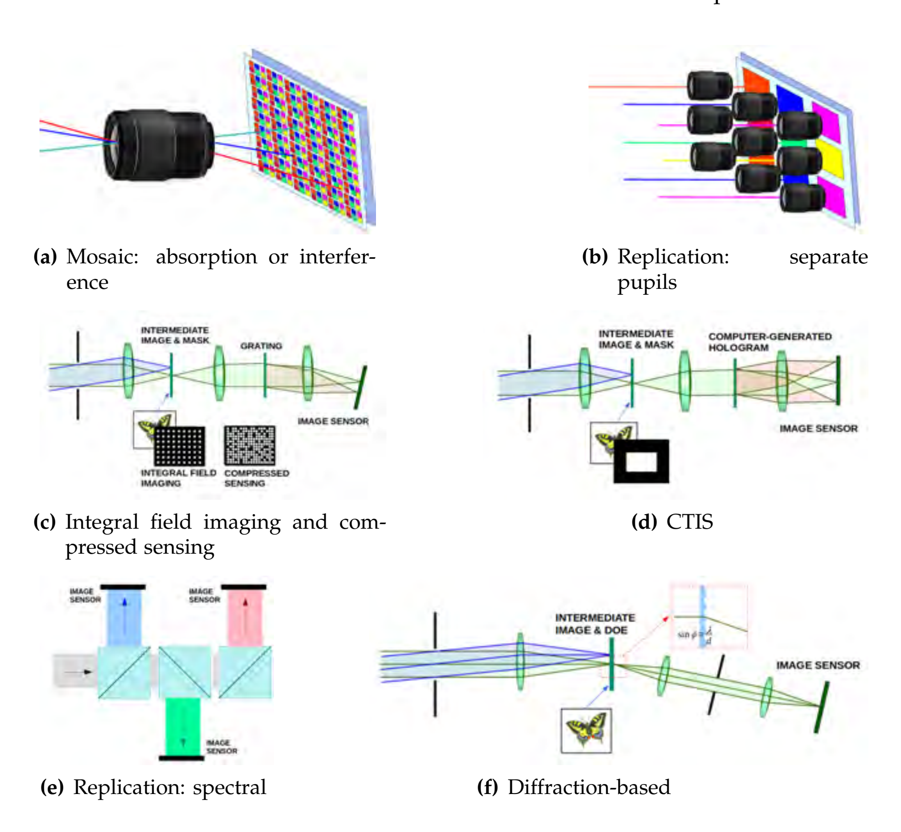
In general one can distinguish between snapshot and scanning sensors. Typically, for a small number of channels snapshot sensors are possible whereas for a large number of channels scanning approaches are applied (most often line-by-line imaging, so-called "push-broom imaging"). In this contribution we focus on snapshot imaging. An excellent overview and review is given by Hagen et al. in [1] and in the following we only will mention the main methods without going into detail about all possible sub-variants. Fig. 1 shows the basic sensing principles that are employed.

Most often, mosaics of absorption-based filters are used (as in the traditional Bayer pattern). This approach has a lot of advantages and is very cheap in mass production.

If more and narrower channels are desired, interference-based filters are employed [2]. Of course, the usable light per spatio-spectral sampling element is proportional to the spectral bandwidth and anti-proportional to the number of spectral channels if there is no spectral overlap (compare section 3). Therefore, for most applications one has to find a trade-off between spatio-spectral sampling and signal-to-noise ratio.

Anyway, disadvantages when using mosaics of dielectric filter have to be kept in mind. Homogeneous manufacturing of areas of such mosaics is complicated and expensive and the spectral response of a filter depends on the angle of incidence of the light (and neighbouring pixels). Therefore, image-sided telecentricity is advantageous. Anyway a thorough calibration of the sensor, ideally for every pixel, is necessary if really sensing with accurate spectral resolution is desired [2].

A variation of the standard mosaic approach is to use image replication. In this case for each of the replicated images an individual filter is employed. Filter manufacturing becomes easier but image replication has to be introduced. Most easily this can be realized macroscopicaly by just using several cameras side-by-side, each one equipped with one individual filter. However, for three-dimensional scenes there is a par-



(a) Mosaic: absorption or interference

(b) Replication: separate pupils

(c) Integral field imaging and compressed sensing

(d) CTIS

(e) Replication: spectral

(f) Diffraction-based

Figure 1: Basic sensing principles for snapshot multi-spectral imaging.

allax between the individual images that should be somehow corrected by image post-processing or otherwise leads to errors. The parallax error is proportional to the separation of the the entrance pupils of the individual image channels. Therefore, miniaturization is advantageous, leading e.g. to approaches like the one described by Hubold et al [3].

A classic alternative is to use the same entrance pupil and to split the image by dichroic beam splitters. This has been used a lot in commercial RGB color cameras because the light efficiency can be improved by that approach, of course at the cost of the need for multiple image sensors and their alignment.

If extensive post-processing is possible, so-called "computed tomography imaging spectroscopy" (CTIS) is an interesting option [4]. The light is diffracted by a computer-generated hologram in multiple diffraction orders that lead to separated copies of the scene on the image sensor. Each copy consists — again — of copies, one for each wavelength. On the sensor one obtains an overlap of all these wavelength separated copies and in the post-processing one tries to reconstruct the original multi-spectral information. Fast implementation is possible using neural networks [5].

The integral field approach uses spatial sampling with a pinhole array in combination with an imaging system with strong (lateral) chromatic aberration to obtain a spectrum for each of the sample points.

Now, if we open more pinholes the naive (and robust) sampling and dispersion approach will fail and we — again — will have overlap of information on the image sensor and some kind of reconstruction to obtain the spatially resolved spectral information is needed. Such approaches are typically denoted as "compressed sensing".

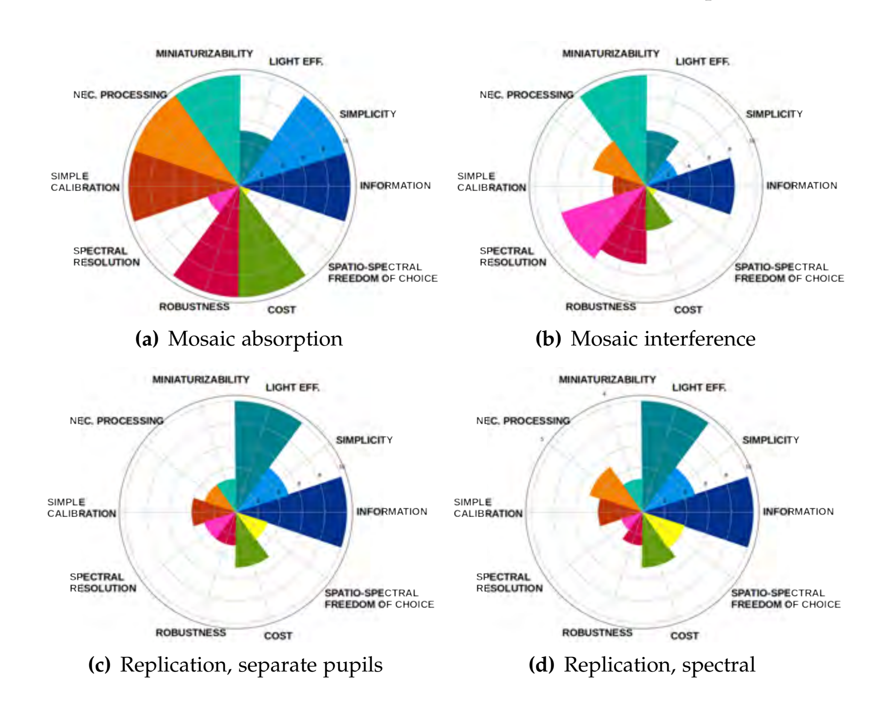
Fig. 2 and 2 show a qualitative comparison of the different sensing principles with respect to the key parameters of a snapshot hyperspectral sensor.

2 Diffraction-based multispectral sensor

One of the main disadvantages of mosaic-based multispectral imaging is the costly and difficult manufacturing of the mosaic filter. For high volume applications, of course, this is not an issue and such filters can be cheaply manufactured. But if specialized areas are to be realized, the initial development cost would be huge.

In Fig. 4 we show an alternative solution that uses diffraction instead of interference or absorption-based filters. It becomes possible to realize arbitrary spatio-spectral patterns by manufacturing a corresponding diffractive optical element. Such manufacturing is possible at rather manageable cost by several companies and universities.

The DOE is located in an intermediate image plane and deflects the light dependent on the wavelength. For understanding the working principle it is beneficial to first assume image-sided telecentricity of the first imaging stage and one large grating with constant grating pe-



(a) Mosaic absorption

(b) Mosaic interference

(c) Replication, separate pupils

(d) Replication, spectral

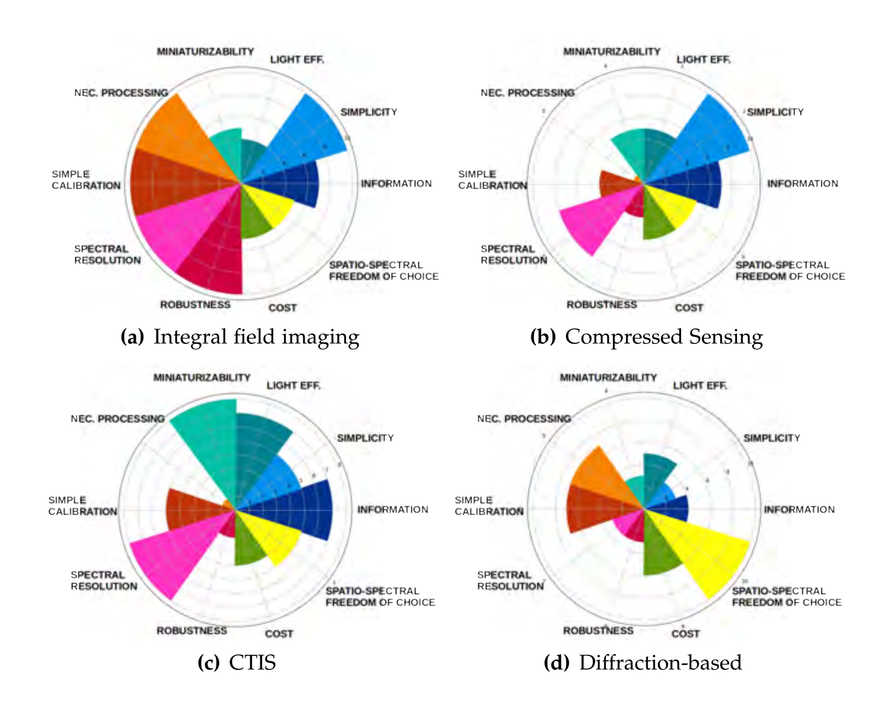
Figure 2: Qualitative comparison of different snapshot multi-spectral imaging approaches (large amplitudes are advantageous). Part 1

riod as the DOE. Due to the telecentricity, the chief rays will arrive at the same angle on the DOE and will be deflected according to their wavelengths.

Different wavelengths then will hit the filter plane (actually the Fourier plane of the second imaging) at different locations and we can, therefore, let a certain spectrum pass the filter by using an appropriate iris. The rest of the second imaging system refocuses the light onto the monochrome (or color, if we want to combine with absorption-based filtering) image sensor.

By that approach we could make a sensor having a certain spectral response but only one channel.

However, we can now replace the simple grating with a more complicated diffracting structure. E.g. we can use different micro-gratings



(a) Integral field imaging **(b)** Compressed Sensing

(c) CTIS **(d)** Diffraction-based

Figure 3: Qualitative comparison of different snapshot multi-spectral imaging approaches (large amplitudes are advantageous). Part 2

with different grating periods at different spatial locations in the intermediate image plane. We choose the periods of the gratings such that a certain wavelength will be deflected in the appropriate way so that it will pass the iris. Each "pixel" in the intermediate image will then consist of a micrograting and the grating period determines which wavelength will pass the iris.

Arbitrary spatio-spectral patterns can be realized by this kind of "grating-mosaic" and one can even realize complex spectra at one point by replacing a micro-grating with a more complex "computer-generated hologram".

Unfortunately, the spectral resolution is strongly coupled with the spatial resolution because the filter acts as a spectral filter *and* the aperture stop of the imaging at the same time. If we use a small hole as the

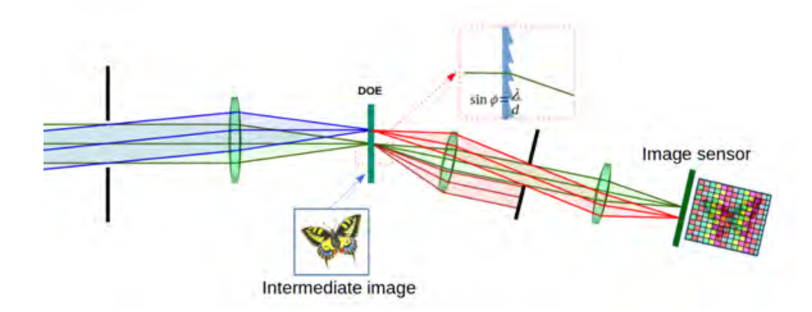


Figure 4: Principle of the diffraction based multi-spectral sensor.

iris, the spectral resolution $\Delta\lambda$ increases but the resolution according to Rayleigh Δr decreases. In [6] we derived the following uncertainty relation, which strongly depends on the minimal grating period d that can be manufactured (Δr is given in the intermediate image plane):

$$\Delta\lambda \cdot \Delta r \geq \lambda \cdot d \tag{1}$$

For a given minimum critical dimension of the DOE manufacturing d and a given size of image (and intermediate image) we will obtain a certain maximum number of resolution cells with a certain spectral bandwidth. This corresponds to the information that can be captured.

With the intermediate image size of $w \times h$ the information is given by

$$\Omega \approx \frac{w\,h}{\Delta r^2} \cdot \frac{\Lambda}{\Delta\lambda} \leq \frac{w\,h\Lambda\Delta\lambda}{\lambda^2\,d^2} \tag{2}$$

if the whole usable spectral range is denoted by Λ.

For an intermediate image with 20 mm $\times$ 20 mm, a minimum grating period of $d = 2\mu m$, a spectral bandwidth of 300 nm and a spectral resolution of 50 nm we obtain $\Omega \approx 5 \cdot 10^6$.

The less spectral channels we use, the larger the overall information that can be captured.

In Fig. 5 we show an example of a measurement with a 7 channel sensor where the DOE consists of stripes. Such an arrangement is es-



Figure 5: Signal on the image sensor during narrow-banded illumination of a USAF target. The USAF target was illuminated in transmission with a central wavelength of 632 nm. The half-width of the illumination spectrum was 10 nm.

pecially useful for detecting small shifts in wavelength. In the shown example spectral shift of 0.5 nm can be measured.

3 Light efficiency

Apart from the quite obvious parameters spatial and spectral resolution, the light efficiency is also very important. Good light efficiency allows one to use larger F-numbers or shorter exposure times at the same signal-to noise ratio.

We want to compare the different multispectral snapshot technologies according to the light efficiency. The baseline is a monochrome image sensor without any spectral channels.

The conventional absorption- or diffraction-based mosaic filter will

be reduced the light efficiency by a factor

$$f_1 = \frac{\Delta\lambda}{\Lambda} \tag{3}$$

Beware that this is not the same than the number of spectral channels. It is advantageous most of the time to have a good light efficiency by using strongly overlapping filters. This is rarely done for commercial sensors but the standard in biology (compare e.g. the spectral responses of cones in the human eye). For classification purposes it is indeed often useful to employ overlapping channels and even simple processing can be used to classify based on spectral information.

In the human visual system, e.g. differences between the red and the green channel are “computed”. The difference signal varies strongly with the spectrum of the input light if it lies in the overlapping region. Therefore, humans are extraordinary good in discerning green-yellowish colors. Obviously, object classification performance as well as light efficiency for ordinary scences is quite good.

The integral field imaging approach uses an amplitude mask in the intermediate image plane [7]. There is no spectral loss of light but, of course, the mask spatially filters and thereby eliminates a lot of photons. The separation of the individual pinholes should be at least N times larger than the diameter of the pinhole if we want to have N separated spectral channels. The associated loss is

$$f_2 = \frac{1}{N} \tag{4}$$

But again we could allow some kind of spectral overlap.

CTIS avoids the use of filtering at all. All incoming photons in principle (we neglect practical issues like the diffraction efficiency of the employed hologram) will arrive at the image sensor. However, it is not clear how to really compare with the filter-based pattern. Due to the overlapping of information a reconstruction step is necessary and at this stage noise might be amplified and artifacts might be introduced. Therefore, the really useful light efficiency is not 100% ($f = 1$). The overall noise is also increased because readout noise, quantization noise and fixed-pattern noise contributions will increase due to the effectively increased number of pixels that are exposed.

Compressed sensing in a snapshot approach lies somewhere between integral field imaging and CTIS. Again, information overlap will occur and, therefore, reconstruction is necessary. However with less information overlap compared to CTIS and less light reduction compared to integral field imaging.

The newly proposed diffraction based approach looses the light at the central iris. And the associated loss is simply again the same as with the integral field imaging approach f_2 if the individual channels are to be spectraly seperated. However, the approach might be worse because the F-number of the first image stage again is coupled with the spectral resolution: a large ray bundle would lead to bad spatial resolution. If one wants to achieve more spatial information with the same spectral behavior one has to increase the size of the intermediate image and as a result the whole setup becomes larger.

The conclusion is: All the approaches lead to more or less the same effective loss of light. The higher the spectral resolution (spectral half width of the channels) is chosen, the more loss is introduced.

One should carefully rethink if high spectral resolution anyway is necessary because overlapping channels are a good thing for a lot of applications.

4 Conclusion

The proposed sensor has the same light efficiency than other well known snapshot multispectral sensing principles. The main advantage is that one can freely chose spatial and spectral resolution at each position of the scene and that even more complex spectral responses can be easily realized using standard diffractive optics manufacturing.

However, spectral and spatial resolution are coupled by an uncertainty relation and also the F-number is coupled to the spectral resolution. In addition, an intermediate image is necessary. In practice this leads to increased space requirements for the sensor.

We thank the German ministry for education and research (BMBF) for financial support under the grant 13N15165 and Simon Amann for fruitful discussion on CTIS.

References

1. N. A. Hagen and M. W. Kudenov, “Review of snapshot spectral imaging technologies,” *Optical Engineering*, vol. 52, no. 9, p. 090901, 2013.
2. R. Hahn, F.-E. Hämmerling, T. Haist, D. Fleischle, O. Schwanke, O. Hauler, K. Rebner, M. Brecht, and W. Osten, “Detailed characterization of a mosaic based hyperspectral snapshot imager,” *Optical Engineering*, vol. 59, no. 12, p. 125102, 2020.
3. M. Hubold, R. Berlich, C. Gassner, R. Brüning, and R. Brunner, “Ultra-compact micro-optical system for multispectral imaging,” in *MOEMS and Miniaturized Systems XVII*, vol. 10545. SPIE, 2018, pp. 206–213.
4. T. Okamoto and I. Yamaguchi, “Simultaneous acquisition of spectral image information,” *Opt. Lett.*, vol. 16, no. 16, pp. 1277–1279, Aug 1991.
5. M. Zimmermann, S. Amann, M. Mel, T. Haist, and A. Gatto, “Deep learning-based hyperspectral image reconstruction from emulated and real computed tomography imaging spectrometer data,” *Optical Engineering*, vol. 61, no. 5, p. 053103, 2022.
6. R. Hahn, T. Haist, K. Michel, and W. Osten, “Diffraction-based hyperspectral snapshot imager,” *Optical Engineering*, vol. 61, no. 1, p. 015106, 2022.
7. R. Bacon, G. Adam, A. Baranne, G. Courtes, D. Dubet, J. Dubois, E. Emsellem, P. Ferruit, Y. Georgelin, G. Monnet *et al.*, “3d spectrography at high spatial resolution. i. concept and realization of the integral field spectrograph tiger.” *Astronomy and Astrophysics Supplement Series*, vol. 113, p. 347, 1995.

Blurred resolution enhancement by graphene nanoplates

Laura Carrilero[1], José R. Castro[1], Sandra Pérez[1], Tomás Belenguer[2], and Félix Salazar[1]

[1] ETSIME (UPM),
C/ Ríos Rosas 21, 28003 Madrid
[2] LINES (INTA),
Torrejón de Ardoz, 28850 Madrid

Abstract Imaging through turbid media leads to a great loss of information decreasing the image quality. In this work we try to palliate this problem by adding an absorbent to the medium, eliminating part of the scattered radiation responsible for the turbidity. This research work is preceded by the demonstration of the effectiveness of black carbon powder as an absorbent, leading to improved quality images [1,2]. With this aim, we use graphene nanoplates as an absorbent and compare the results with black carbon powder in order to study the possible improvement.

Keywords Vision, absorption, scattering, turbid media

1 Introduction

When a medium is interposed between an object and the detection system, there is a loss of quality of the transmitted image due to the light behavior through the medium. The transparency property of a system affects how the light behaves passing through it. For instance, translucent materials, such as diffusive media, allow light to pass through them, but it suffers changes. Some photons pass through the body and reach the detector without alterations (ballistic photons), some fail to pass through it and are retained in the medium (absorbed photons) and others suffers changes in its trajectory (scattered photons), not allowing a clear vision, since they arrive the detector in a random manner. Adding an absorber to the medium can improve the image quality

since it has more chances to absorb the scattered photons due to its longer path than the ballistic ones, hence part of the scattered radiation will be eliminated before reaching the detector.

Consequently, numerous studies have been carried out in these media for several years, giving rise to certain mathematically complex techniques such as stellar interferometry, inverse scattering, or fluorescence, among many others, trying to solve this problem. Regarding fluorescence, it is worth mentioning a test performed at the end of the 20th century, where a technique to improve the image quality of an object hidden by a diffuse medium combining fluorescence and absorption was tested, in it was shown that the image quality could be further improved by absorption, selecting the spectral range of the fluorescence light that is highly absorbed by the medium [3]. In addition, at the end of the 20th century, a new technique, simpler to perform, was introduced to improve vision through a random medium with high diffusion by using the absorption present in the medium. It was proven that absorption reduces the intensity of scattered light, that generates the image noise, below the intensity of the ballistic signal, which forms the image. This reduction in the signal-to-noise ratio allows to see through a diffusive medium that would be opaque without the presence of absorption [4]. Another test at this time showed that by using the absorption method to improve image quality in turbid media, the received energy decreases, but so does the path of the photons arriving at the detector, meaning that more scattered photons are absorbed, with a higher trajectory, than ballistic photons. In addition, it showed that the results obtained were similar to those achieved with the time-gating technique, which is more complex and expensive. This method is the most widely used for breast imaging. This last trial was performed to contemplate a new technique in medicine to detect breast tumors [5]. Gradually, the applications of this methodology have grown, reaching the military industry [6], and the astronomy [7].

The basic methodology we have used is similar that developed in [1]. In this work, the improvement of image quality is studied using black carbon powder as an absorber in two different scattering media, one consisting of zinc oxide nanoparticles, and the other of polystyrene nanoparticles. This last technique is the one of interest in the present investigation and on which the study has been based. For this purpose, we have made a series of samples and tested them in the labo-

ratory for subsequent analysis in Matlab using the SSIM function. In 2018 another paper was published following the research line of the 2016 [2]. In this article the influence of the wavelength of the incident light on the image enhancement is studied. Due to the angular distribution of the scattering depends on the size of the scatterers with respect to the wavelength of the incident light, they determine a new approach to image enhancement, selecting the appropriate wavelength range [2]. The most recent paper we have found is from 2019, in which authors analyze the absorption-scattering coupling and its impact on haze in random media. They also introduce the haze-absorption sensitivity spectrum which quantifies the capacity of absorption-induced haze supression [8].

It is also worth mentioning other interesting papers about absorption, scattering and turbid media, using other techniques and approaches [9–19].

Taking into account the aforementioned investigations, the aim of this paper is to compare the image enhancement achieved by graphene and black carbon powder as absorbers, and to study the influence of incident light by performing the experiments using white and red light.

2 Theory

To understand the absorption phenomenon, light must be understood as a corpuscle, quantized, with discrete values of energy. Absorption occurs when an electron is excited by a photon. Electrons occupy orbitals separated from each other by discrete amounsts of energy, in which the number of electrons is limited by the Pauli exclusion principle. When excited, the electron will move to a higher energy level, absorbing the energy and leaving a hole in its original position.

Imaging through absorbers leads to a loss of brightness since not all ballistic photons reach the detector. In the context of the image vision, scattering occurs, for instance, when a particulate system is interposed between the object and the detection system, such as turbid media. The rays emitted by the object are obstructed by the particles in the medium, deflecting their path.

Scattering depends on the particle size. We can distinguish two models within the context of our work: the Rayleigh and the Mie regimes.

When the particle size is much smaller than the wavelength of the incident light, we are in the Rayleigh case, while if the particle diameter is of the order or larger than the wavelength of the incident light, it is Mie scattering.

Imaging through scatterers generates a loss of quality of the trasmitted image, since the scatterers deflect the photons that arrive disorderly at the detector, resulting in blurred images. Therefore, in turbid media we can distinguish three types of photons: the ballistic photons that form the image, arriving in an orderly manner at the detector; the scattered photons, which generate blurred images because their trajectory has been altered and they arrive randomly at the camera; and the absorbed photons, which do not reach the detector, causing a loss of intensity.

3 Methodology

The procedure we have followed to perform the experiments its shown in the diagram below 2.

It has consisted of, first of all, the preparation of the samples, in which we have used graphene and black carbon powder as absorbers (both separately), polystyrene nanospheres as diffusers and distilled water as the matrix medium. We tested four different solutions, gradually increasing the amount of diffuser, with concentrations of 30, 50, 70 and 100 µl in 10 ml of distilled water, and absorbers concentrations of 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 1.3 and 3.3 mg for the first three solutions and 0.5 and 1.3 for the last one.

Afterwards, we introduce the samples into the cuvette of the optical system for imaging. The imaging setup we use consisted of a CMOS camera, a 1951 US Air Force resolution target as the object, a biconvex convergent type lens, a rectangular glass cuvette to place the samples in between the camera and the object. As radiation source, we used an incoherent white led light with a 650 nm bandpass filter, and without.

Once the images have been taken, to compare and evaluate their quality, first, we must select those that are comparable with each other, for what we use the signal-to-noise ratio following three different methods depending on the application. To calculate the signal-to-noise ratio

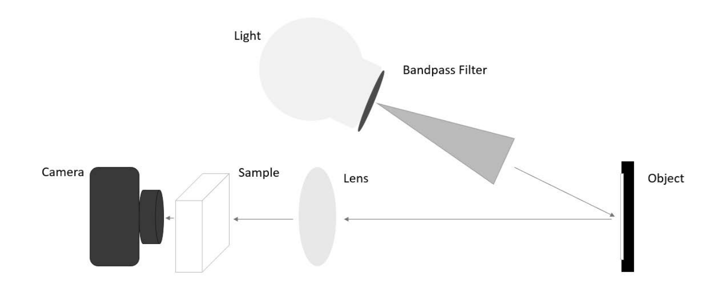


Figure 1: Imaging setup.

of the images we use the following formula:

$$SNR = \frac{\prec I \succ}{\sqrt{\prec I^2 \succ - \prec I \succ^2}}$$

Where, I = Image intensity and $\prec I \succ$= Average image intensity

Here, we present the methods we have used to choose the images based on the application. The resulst vary depending on the method selected.

Method 1: We select the images considering the signal-to-noise ratio of the reference image. This criterion is useful for those applications where the reference image is known, for example, in the geostationary satellite case.

Method 2: Considering the exposure time of the reference image we select the disturbed one and, depending on its signal-to-noise ratio, we choose the images with an absorber. This criterion is useful for images whose damage degree is such that it is not possible to return to the reference one, and therefore, it is necessary to work on the disturbed image. For instance, in optical space elements that have suffered such a deterioration that you cannot return to their initial conditions, and therefore it is required to work with the deteriorated image.

Method 3: Considering the exposure time of the reference image we select the same disturbed one. Then, to select the images with an absorber we vary the exposure time seeking to return to the signal-to-noise ratio conditions of the reference. This criterion is useful for

applications like method 1 but when the disturbance appears instantaneously.

Once we have selected the images to evaluate them we use the structural similarity index (SSIM) in Matlab.

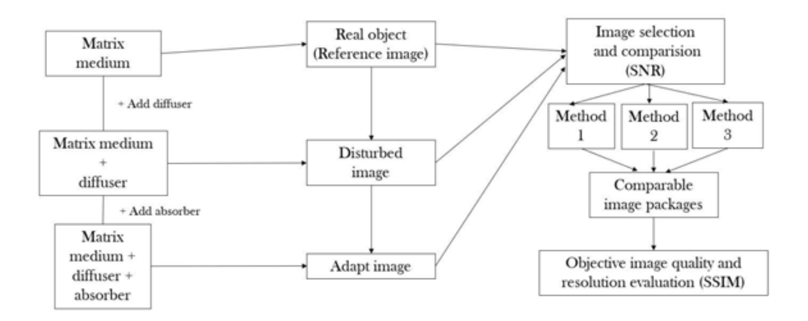


Figure 2: Procedure diagram.

SSIM quantifies the similarity of an image regarding the reference one, taking into consideration the structure, contrast, and illuminance of the images (x,y), as we can see below [20,21].

1) Iluminance comparison:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},$$

where $C_1 = (0,01 \cdot 2^{bitsperpixel} - 1)^2$

2) Contrast comparison:

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

being $C_2 = (0,03 \cdot 2^{bitsperpixel} - 1)^2$.

3) Structure comparison:

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Here, $\mu_x, \mu_y, \sigma_x, \sigma_y$ and σ_{xy} are the local means, standard deviations and cross-covariance. From these quantities may be demonstrated that the SSIM index is:

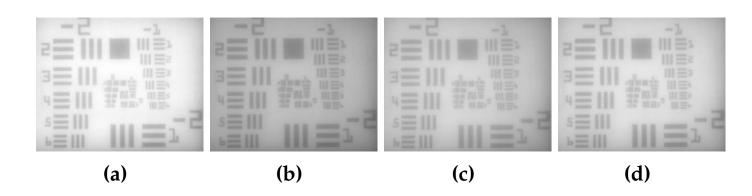
$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

4 Experimental results

In this section we show the most important results obtained. We have compiled those experiments in which an improvement of graphene over graphite is detected, for each method explained above with red and white light sources. First we show the results for red light for each technique, and then those obtained with white light.

As we can observe, both numerically by means of SSIM and visually, the fourth image on the right, adapted with graphene, is the one that most resembles the reference picture, improving with respect to the image perturbed with the diffuser and the image adapted with graphite (Figures 3 to 8).

For the image series from 3 to 5 with red light we observe that the SSIM values achieved for the adapted versus perturbed images present larger differences than for the white light case (Figures 6 to 8), especially for method 1.



(a) **(b)** **(c)** **(d)**

Figure 3: Results of the method 1 for red light. (a) Reference image 10ml distilled water. (b) Disturbed image 30 μl polystyrene, SSIM=0.5497. (c) Image with 0.4 mg graphite, SSIM=0.6818. (d) Image with 0.4 mg graphene, SSIM=0.7849.

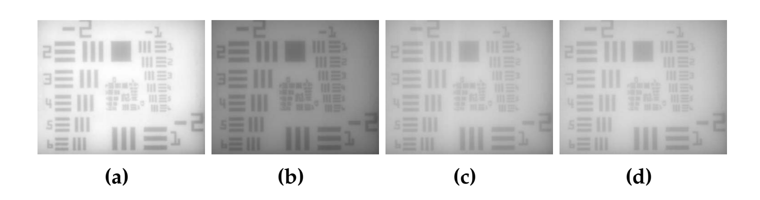


Figure 4: Results of the method 2 for red light. (a) Reference image 10ml distilled water. (b) Disturbed image 50 μl polystyrene, SSIM=0.4105. (c) Image with 3.3 mg graphite, SSIM=0.6629. (d) Image with 3.3 mg graphene, SSIM=0.7107.

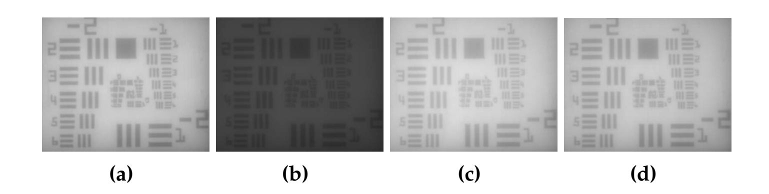


Figure 5: Results of the method 3 for red light. (a) Reference image 10ml distilled water. (b) Disturbed image 70 μl polystyrene, SSIM=0.5868. (c) Image with 0.4 mg graphite, SSIM=0.9717. (d) Image with 0.4 mg graphene, SSIM=0.9968.

The results obtained using white light for the three methods are shown below.

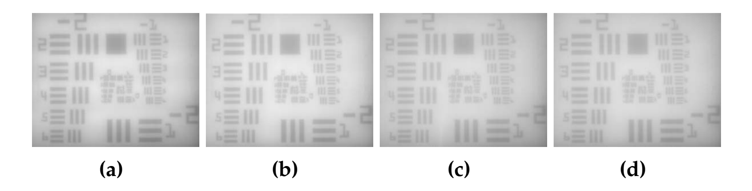


Figure 6: Results of the method 1 for white light. (a) Reference image 10ml distilled water. (b) Disturbed image 70 μl polystyrene, SSIM= 0.9762. (c)Image with 0,6 mg graphite, SSIM=0.9918. (d) Image with 0.6 mg graphene, SSIM=0.9920.

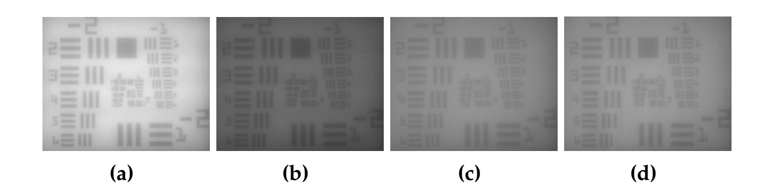


Figure 7: Results of the method 2 for white light. (a) Reference image 10ml distilled water. (b) Disturbed image 50 μl polystyrene, SSIM= 0.5815. (c) Image with 3.3 mg graphite, SSIM=0.7384. (d) Image with 3.3 mg graphene, SSIM=0.7736.

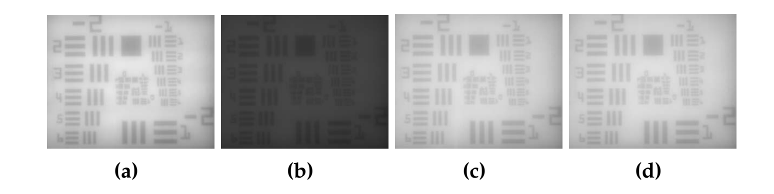


Figure 8: Results of the method 3 for white light. (a) Reference image 10ml distilled water. (b) Disturbed image 70 μl polystyrene, SSIM= 0.4525. (c) Image with 0.6 mg graphite, SSIM=0.9918. (d) Image with 0.6 mg graphene, SSIM=0.9920.

5 Conclusion

In this paper we have presented the image enhancement by the absorption technique using graphene as an absorber. Likewise, a comparison between the enhancement obtained by graphene and graphite, and white and red light, has been made.

We have found that, in most cases, for the type 2 suspension and red light, the concentration at which the best SSIM values are achieved for graphene is 0.4 mg. We have encountered that in the case of vision loss due to image intensity saturation, there is a generalized improvement when introducing both, polystyrene nanospheres and the two absorbers. We expected significant results in which the enhancement would be visible to the naked eye for any of the three methods, however, for method 1 the improvements are practically negligible, being visibly unnoticeable. Also, we found the most important results for method 3, and the least remarkable for method 1. We found more significant results with red light rather than with white light. In addition, white light saturates sooner.

References

1. M. Tanzid, N. J. Hogan, A. Sobhani, H. Robatjazi, A. K. Pediredla, A. Samaniego, A. Veeraraghavan, and N. J. Halas, "Absorption-induced image resolution enhancement in scattering media," *ACS Photonics*, vol. 3, no. 10, pp. 1787–1793, 2016.
2. M. Tanzid, N. J. Hogan, H. Robatjazi, A. Veeraraghavan, and N. J. Halas, "Absorption-enhanced imaging through scattering media using carbon black nano-particles: From visible to near infrared wavelengths," *Journal of Optics*, vol. 20, no. 5, p. 054001, 2018.
3. K. Yoo, Z.-W. Zang, S. A. Ahmed, and R. Alfano, "Imaging objects hidden in scattering media using a fluorescence–absorption technique," *Optics letters*, vol. 16, no. 16, pp. 1252–1254, 1991.
4. K. Yoo, F. Liu, and R. Alfano, "Imaging through a scattering wall using absorption," *Optics letters*, vol. 16, no. 14, pp. 1068–1070, 1991.
5. D. Contini, H. Liszka, A. Sassaroli, and G. Zaccanti, "Imaging of highly turbid media by the absorption method," *Applied optics*, vol. 35, no. 13, pp. 2315–2324, 1996.

6. P. B. Schwering, R. A. Kemp, and K. Schutte, "Image enhancement technology research for army applications," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXIV*, vol. 8706. SPIE, 2013, pp. 198–208.
7. R. Tyson, "Principles of adaptive optics third edition," 2010.
8. L. Zhao, M. Blackman, L. Zhang, B. Bhatia, A. Leroy, E. Strobach, and E. N. Wang, "Plasmonic absorption-induced haze suppression in random scattering media," *Applied Physics Letters*, vol. 114, no. 25, p. 251102, 2019.
9. J. J. Dolne, K. M. Yoo, F. Liu, and R. R. Alfano, "Continuous wave near infrared fourier space and absorption imaging through random scattering media," in *Advances in Laser and Light Spectroscopy to Diagnose Cancer and Other Diseases*, vol. 2135. SPIE, 1994, pp. 209–212.
10. S. F. Liew, S. M. Popoff, A. P. Mosk, W. L. Vos, and H. Cao, "Transmission channels for light in absorbing random media: from diffusive to ballistic-like transport," *Physical Review B*, vol. 89, no. 22, p. 224202, 2014.
11. S. F. Liew and H. Cao, "Modification of light transmission channels by inhomogeneous absorption in random media," *Optics express*, vol. 23, no. 9, pp. 11 043–11 053, 2015.
12. M. Tanzid, A. Sobhani, C. J. DeSantis, Y. Cui, N. J. Hogan, A. Samaniego, A. Veeraraghavan, and N. J. Halas, "Imaging through plasmonic nanoparticles," *Proceedings of the National Academy of Sciences*, vol. 113, no. 20, pp. 5558–5563, 2016.
13. X. L. Deán-Ben, H. Estrada, A. Özbek, and D. Razansky, "Controlling the light distribution through turbid media with wavefront shaping based on volumetric optoacoustic feedback," in *Adaptive Optics and Wavefront Control for Biological Systems II*, vol. 9717. SPIE, 2016, pp. 198–202.
14. J. A. Carr, M. Aellen, D. Franke, P. T. So, O. T. Bruns, and M. G. Bawendi, "Absorption by water increases fluorescence image contrast of biological tissue in the shortwave infrared," *Proceedings of the National Academy of Sciences*, vol. 115, no. 37, pp. 9080–9085, 2018.
15. M. I. Mishchenko, L. Liu, and J. W. Hovenier, "Effects of absorption on multiple scattering by random particulate media: exact results," *Optics express*, vol. 15, no. 20, pp. 13 182–13 187, 2007.
16. S. A. Ahmed, Z.-W. Zang, K. M. Yoo, M. Ali, and R. Alfano, "Effect of multiple light scattering and self-absorption on the fluorescence and excitation spectra of dyes in random media," *Applied optics*, vol. 33, no. 13, pp. 2746–2750, 1994.
17. W. Liu, Z. Zhou, L. Chen, X. Luo, Y. Liu, X. Chen, and W. Wan, "Imaging through dynamical scattering media by two-photon absorption detectors," *Optics Express*, vol. 29, no. 19, pp. 29 972–29 981, 2021.

18. M. K. Swami, S. Manhas, H. Patel, and P. K. Gupta, "Mueller matrix measurements on absorbing turbid medium," *Applied Optics*, vol. 49, no. 18, pp. 3458–3464, 2010.
19. Z. Feng, T. Tang, T. Wu, X. Yu, Y. Zhang, M. Wang, J. Zheng, Y. Ying, S. Chen, J. Zhou *et al.*, "Perfecting and extending the near-infrared imaging window," *Light: Science & Applications*, vol. 10, no. 1, pp. 1–18, 2021.
20. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
21. A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

Optimal human labelling for anomaly detection in industrial inspection

Tim Zander,[1,2] Ziyan Pan,[1] Pascal Birnstill[2], and Juergen Beyerer[1,2]

[1] Institut für Anthropomatik und Robotik, Lehrstuhl für Interaktive Echtzeitsysteme, Karlsruher Institut für Technologie (KIT)
[2] Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB)

Abstract Anomaly detection with machine learning in industrial inspection systems for manufactured products relies on labelled data. This rises the question how the labelling by humans should be conducted. We consider the case where we want to optimise the cost of the combined inspection process done by humans and an algorithm. This also influences the combined performance of the trained model as well as the knowledge of the performance of this model. We focus on so called one-class classification problem models which produce a continuous outlier score. We establish some cost model for human and machine combined inspection of samples. We then discuss in this cost model how to select two optimal boundaries of the outlier score where in between these two boundaries human inspection takes place. We also frame this established knowledge into an applicable algorithm.

Keywords Mathematical methods and models, artificial intelligence and machine learning, quality control

1 Introduction

The detection of non-common patterns in a batch of samples is a strong point of human visual cognition. Still there are many known limitations to human visual inspection as well as cost issues in real world production systems. The training of machine learning models for anomaly detection of industrial inspection problems is often done as a one-class classification problem where only good samples are presented to the

algorithm. The background for this is that it is in general easy to acquire good samples but difficult and expensive to find anomalous samples. A dataset for benchmarking this type of algorithm is the MVTec-dataset [1] [2]. The best performing model[3] on this dataset is *"Patchcore"* [3]. For a given picture sample a *"Patchcore"*-model after training produces an outlier score together with a heat map on the likelihood of being an anomalous area. This is done by performing outlier-detection on the deep-features of a pretrained neural network of the images. The cutoff values for an anomaly in the outlier score of *"Patchcore"* are optimised in the paper by finding the cutoff-value with the highest F1-score. This already assumes that there are known outliers which are potentially very costly to acquire. Although we think of models designed for the MVTec dataset like *"Patchcore"* as the main application, our method of finding two boundaries for the outlier score, where in-between human inspection will take place, will work for any model of an one-class classification problem [4] with a continuous score.

More precisely, in this paper we formulate the problem of optimal usage of human inspection after acquiring of initial data for training. For this we assume that there are certain costs for inspection and costs for falsely classified samples. We are not are aware that such a human-in-the-loop machine learning consideration exists in the literature, although more generic considerations about iterative machine teaching and active learning can be found in [5]. A similar process by giving the human some sort of optimal presentation of data for labelling was done in [6]. However, this method is not applicable for the one-class outlier classification problems on images we consider here. In [7] it is shown, that for one-class classification models one can train an additional model on the bad samples and use a combined score on the good and bad sample models to find the most promising new samples for labelling. The authors show that using one of their active learning methods one can achieve faster convergence and better overall performance of the model. We refer to Munro's book [8] for a general overview of human-in-the-loop machine learning.

Another important concept which we will discuss and use is that of probabilistic classifiers. Probabilistic classifiers are classifiers which output a probability distribution on the target classes instead of just a

[3] https://paperswithcode.com/sota/anomaly-detection-on-mvtec-ad

score. Model calibration is a technique which achieves that a classifier will have a probabilistic output [9] [10]. A calibrated one-class classifier will give out a probability p which will represent the probability of being in the one class. In safety-critical applications it is important to have an idea of uncertainty of the model. Hence a probabilistic output is of great help with regard to such problems. Even in situations which are just cost-critical we will show that we can exploit having an uncertainty estimate of the classifier for a given sample to make better decisions.

2 Model

In this section we will describe the necessary pre-conditions and cost assumptions. Further we describe how, after initial training of our one-class classifier, we can establish our first optimal boundaries. We do describe multiple alternatives here. Then we pass on to acquiring more knowledge about the outliers we will encounter and their outlier scores. This will then be used to establish optimal decisions for the cutoff parameters of human inspection in the sense of our pre-made cost assumptions.

2.1 Pre-conditions

First we introduce a few more preliminary and formal assumptions and notations. We assume that there exists a set of images or more general data I which each have a hidden label $\{0,1\}$ where images with label 0 are good samples and images with label 1 are anomalous samples. We will observe these samples in some process such as an industrial inspection task one after another. For our cost considerations we assume that the process of labelling a sample by a human has a cost c_l associated with it. Further we assume that human labelling perfectly assigns the correct label to the data. With N initially labelled data points we train and test a model M which will then produce an outlier-score $M(i) \in \mathbb{R}$ for every (new) image i we observe. We set a lower and upper decision boundary for manual inspection b_l and b_u such that any image i with outlier score $M(i)$, where $b_l < M(i) < b_u$ holds, will be inspected by a human.

2.2 A priori cost and anomalous data

For our cost considerations we further assume that there is a known (possibly non-linear[4]) cost-function C_f such that the absolute cost of missed outliers can be calculated as $C_f(\mathbf{FOR}) \cdot K$ where **FOR** is the false omission rate, i.e. the percentage of anomalies in the accepted samples, and K the absolute number of accepted samples. The cost of false positive samples are associated with a cost per sample of c_r. This could be for example lost revenue and disposal costs of an unnecessarily disposed sample of good state.

2.3 Initial cut-off boundaries

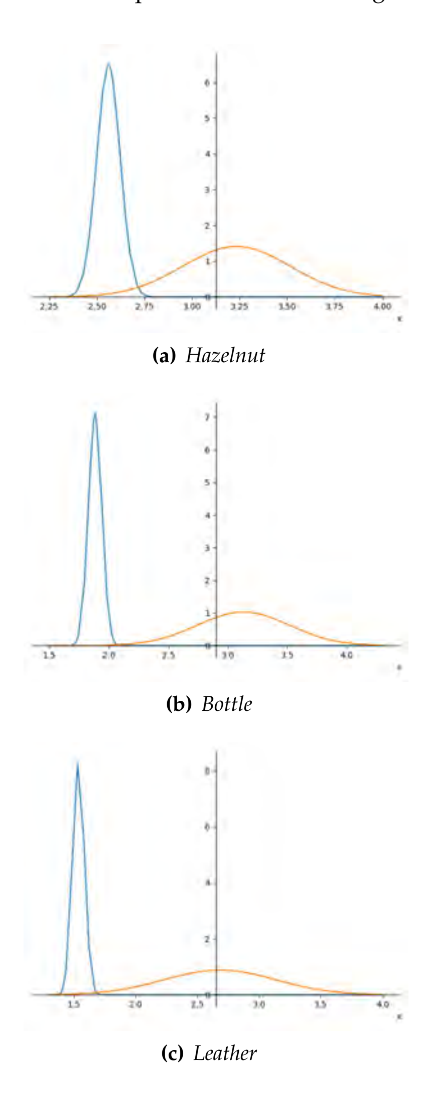
We assume now that the initial sampling and labelling of data D and the training of a model M is conducted. We update our initial belief p_o of the outlier percentage by taking the percentage of outliers in the sampled D into account. We are now interested in finding optimal cutoff parameters b_l, b_u in this stage. We discuss multiple alternatives now.

A priori anomaly distribution

In the first case we assume that the distribution of the outlier score of samples with label 0 and also of the samples with label 1 is both Gaussian[5]. For the good samples we can directly estimate this distribution. We get some distribution g_g with mean μ_g and variance σ_g. For the bad samples we also get some Gaussian distribution g_b. In the case where there are no bad samples available, we take some initial belief about the distribution, which we could take from former observations such as the MVTec dataset or a similar product line (see Figure 1), as our distribution. We can find the optimal parameters b_l, b_u in terms of cost. In order to find these parameters one would minimise Equation 1 of Section 2.4.

[4] One reason for non-linearity could be reputation costs, i.e., due to network effects reputation falls non-linearly with increasing fault-rate.

[5] A non-Gaussian distribution could also easily be considered here.



(a) *Hazelnut*

(b) *Bottle*

(c) *Leather*

Figure 1: These are the Gaussian distributions of anomaly scores for different items from the MVTec Dataset. Blue represents the good sample distribution and red represents the bad sample distribution. The model where the anomaly score stems from was Patchcore [3] and it was trained with training sample split of the MVTec dataset. Then the anomaly score output of the trained model on the good and bad samples of the test dataset split was used to find the shown Gaussian distributions. On these data-sets the established model has an AUC-score of 0.9996 for *Hazelnut*, 1.0 for *Bottle* and 1.0 for *Leather* on the test dataset samples.

Optimal cut-off sigma

Another approach would be to omit to define an a priori distribution of g_b and instead take a cutoff parameter x such that any sample with outlier score higher than $\mu_g + x \cdot \sigma_g$ is considered anomalous. The choice of the parameter x can be done as follows. We assume that we cannot inspect every piece which we observe but only some percentage p_i of it. Hence we have to find x in such a way that the expected amount of samples classified as anomalous is at most the amount that can be handled. Hence we have to pick x such that

$$p_i \geq (1 - p_o) \int_{\mu_g + x \cdot \sigma_g}^{\infty} g_g + p_o$$

holds. Note that we omitted the expected false negative classified samples in our considerations, but we assume that this amount is negligibly small. In case there is no sample to classify at the moment we might pick a random sample. In case we acquired enough bad samples we can infer the distribution g_b or update our initial belief about it. More details on the belief update of a Gaussian distribution can be found in [11].

Calibrated output

In some cases the model comes with a calibrated probabilistic output. This roughly means that the output value of the model $M(*)$ is a probability of being an outlier, e.g. we expect to find $q * 100$-many outliers of 100-samples i' with score $M(i') = q$. With such a calibrated model we can directly use the model output as our probability. We will not further assume that our model is calibrated although the following should be straightforward to adapt for directly using this output instead of learning some probability as in the previous paragraph.

Now we have found a priori parameters b_l, b_u or just $b_l (= \mu_g + x \cdot \sigma_g)$. With these we can set up our initial human in the loop process. After some time we will enrich our dataset of labelled pieces and therefore can update our believe about the Gaussian curves g_g, g_b as described in [11] or interfere the distributions g_g, g_b directly from all the gathered data. There is some caveat with the selection of the samples: Because

of our parameters the selection of the samples is biased. This either needs to be corrected through enough random samples or giving the unlabelled data some pseudo label with continuous value greater 0 and smaller than 1. Additionally we could use the gathered data to further improve the model M or respectively re-train a new M with the new data and old data depending on the algorithm in use. In any case we now fix some model M, some p_0 and the Gaussian distributions g_g, g_b associated with it as well as the gathered data. In case we observed and classified a new sample we could continue to do a belief update of our estimated values p_o, g_g and g_b and retrain our model M in order to keep improving it. But we omit such considerations in the rest of the paper.

2.4 Cost-calculation

We calculate the cost associated for some fixed b_l and b_u for the next samples. We expect to see p_o-percent outliers which we have updated from the observations D. Additionally we can calculate the expected percentage that the next sample will be true positive: $\mathbf{TP}(b_l) = p_o \int_{b_l}^{\infty} g_b$, true negative: $\mathbf{TN}(b_u) = (1 - p_o) \int_{-\infty}^{b_u} g_g$, false negative: $\mathbf{FN}(b_l) = p_o \int_{-\infty}^{b_l} g_b$ and false positive: $\mathbf{FP}(b_u) = (1 - p_o) \int_{b_u}^{\infty} g_g$. From this we can calculate the false omission rate $\mathbf{FOR} = \frac{\mathbf{FN}}{\mathbf{FN}+\mathbf{TN}}$. Now for the next sample have the cost function $\mathcal{C}(b_l, b_u)$ defined as follows:

$$
\begin{aligned}
&C_f(\mathbf{FOR}(b_l)) \cdot [\mathbf{TN}(b_u) + \mathbf{FN}(b_l)] + c_r \cdot \mathbf{FP}(b_u) + \\
&c_l \cdot (1 - p_o) \int_{b_l}^{b_u} g_g + c_l \cdot p_o \int_{b_l}^{b_u} g_b.
\end{aligned} \tag{1}
$$

This function is our minimisation target for which we choose b_l and b_u accordingly:

$$
\begin{aligned}
\min_{b_l, b_u} \quad & \mathcal{C}(b_l, b_u, g_b, g_g, p_0) \\
\text{s.t.} \quad & b_l \leq b_u \\
& b_l, b_u \in \overline{\mathbb{R}}
\end{aligned} \tag{2}
$$

where $\overline{\mathbb{R}}$ is the set of the extended real numbers which additionally contains plus and minus infinity, i.e. $\mathbb{R} \cup \{-\infty, +\infty\}$.

In case of a very low outlier rate p_o we can simplify the cost by setting $b_u = \infty$ and the optimisation problem becomes a single variable problem. Often it will be the case that we have a fixed percentage of images, say p_f, which we can inspect due to such things as fixed amount of available human labour. In this case the lower part of the cost function 1 will be replaced by the constraint

$$p_f = (1 - p_o) \int_{b_l}^{b_u} g_g + p_o \int_{b_l}^{b_u} g_b.$$

If we additionally set $b_u = \infty$ we can already find the optimal b_l by just using this constraint. But these considerations are still useful as we can now estimate the cost of our system and further estimate whether it is useful to employ or dismiss a human at a certain cost or estimate the cost saving for a higher or lower rate of inspection of samples.

2.5 Non independence of outlier observations

In the case where we believe there is a non-independence of the series of observed data[6] we could increase the believed percentage of outliers p_o for the next few observed samples after observing an outlier. This ensures that the costs stay optimal for the next observed samples with higher anomaly probability. Note that in more complicated production environments we may observe pieces from multiple different machines. If possible one should keep track of the machines a piece went through to get more individual assessment of the anomalous probabilities.

3 Algorithm

In this section we combine the observations established in the last section into an combined algorithm (see Algorithm 1). As an input to our algorithm there is a one-class classification model M that needs N-many samples for initial training and testing, and there is also a belief about the percentage of outliers p_o in the samples to be observed. Additionally we have the cost function C_f and a real value c_r representing the cost of a false positive sample. Moreover we fix an amount of outliers we want to observe L. The algorithm starts by letting a human

[6] A broken machine could for example produce a sudden stream of defect parts.

label samples till we receive a set D containing N-many labelled samples with label 0. We use this dataset D to train some model M_D which then is used to produce some outlier scores for the test data split of D. This is then used to find more anamalous samples in order to form a probabilistic model by inferring a Guassian curve of the good and a Gaussian curve of the bad samples. With this we are finally able to find the cost optimal parameters b_l and b_u which mark the outlier score interval where human inspection takes place.

Algorithm 1 Find optimal interval for human inspection

1: initialization: p_o, C_f, c_r, c_l, N, L
2: $n \leftarrow 0$
3: **for** $n < N$ **do**
4: wait for next sample s
5: get label $l(s)$ (by human)
6: $n \leftarrow n + 1 - l(s)$
7: $p_o \leftarrow$ belief update through observed $l(s)$
8: **end for**
9: **return** training dataset D , p_0
10: $M_D \leftarrow$train model with D
11: $b_l' \leftarrow$ (see Section 2.3 for possible computations)
12: $k \leftarrow 0$
13: **for** $k < L$ **do**
14: get next sample s
15: **if** $b_l < M_D(s)$ **then**
16: get label $l(s)$ (by human)
17: **end if**
18: $k \leftarrow k + l(s)$
19: $p_o \leftarrow$ belief update through observed $l(s)$
20: **end for**
21: **return** updated dataset D, p_0
22: $g_g, g_b \leftarrow$ interfere Gaussian from data D
23: solve $\min_{b_l,b_u} \mathcal{C}(b_l, b_u, g_b, g_g, p_0)$
24: **return** Model M_D and inspection interval values b_l, b_u

4 Discussion and future work

We establish theory for the cost-optimal selection of samples of one-class classifications models. For this we established a cost-model and showed how to infer probabilistic knowledge of the samples online and offline in order to establish a cost-optimal decision for a human inspection boundary in the outlier score. Moreover, we have merged this into an algorithm which can be applied in production. For now we have not considered the case of retraining the model and we can assume that this will be done occasionally till the economic evaluation stabilises or the performance is satisfactory. Also the problem of a timely dependence of the occurrence of outliers which could stem from faulty machines was discussed. At worst there could be no outlier samples or only a very biased selection of them. A detailed analysis of the practical relevance of this problem could be an interesting topic for future investigation. There could also be potential for future work especially in the case where the one-class problem is a moving target, i.e. the golden sample changes over time. The case for selecting valuable examples for improving the model performance also seems an interesting area not yet considered and will probably require an extra model which is also trained with the outliers. Another not yet used feature is utilising the presentation of anomalous areas on the image for better outlier visualisation for the user decision. There, another optimisation problem arises which is the optimisation of the cutoff parameter for the selection of the anomalous area. A more general question is the question of a good visualisation to improve human performance.

References

1. P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.
2. P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1038–1059, 2021.

3. K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 318–14 328.
4. P. Perera, P. Oza, and V. M. Patel, "One-class classification: A survey," *arXiv preprint arXiv:2101.03064*, 2021.
5. E. Mosqueira-Rey, D. Alonso-Ríos, and A. Baamonde-Lozano, "Integrating iterative machine teaching and active learning into the machine learning loop," *Procedia Computer Science*, vol. 192, pp. 553–562, 2021.
6. C. Chai, L. Cao, G. Li, J. Li, Y. Luo, and S. Madden, "Human-in-the-loop outlier detection," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 19–33.
7. P. Schlachter and B. Yang, "Active learning for one-class classification using two one-class classifiers," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1197–1201.
8. R. Munro, *Human-in-the-loop machine learning*. New York, NY: Manning Publications, Oct. 2021.
9. J. Vaicenavicius, D. Widmann, C. Andersson, F. Lindsten, J. Roll, and T. Schön, "Evaluating model calibration in classification," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 3459–3467. [Online]. Available: https://proceedings.mlr.press/v89/vaicenavicius19a.html
10. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.
11. K. Murphy, "Conjugate bayesian analysis of the gaussian distribution," 11 2007. [Online]. Available: https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf

Quality control of laser welds based on the weld surface and the weld profile

Julia Hartung[1,2], Andreas Jahn[2], and Michael Heizmann[1]

[1] Karlsruhe Institute of Technology, Institute of Industrial Information Technology, Hertzstraße 16, 76187 Karlsruhe, Germany
[2] TRUMPF Laser GmbH, Aichhalder Str. 39, 78713 Schramberg, Germany

Abstract 2D or 3D sensor technology can be used for data acquisition to monitor the weld quality during laser welding. Compared to a 2D camera image, the 3D height data contains additional relevant information for quality inspection. However, the disadvantages are system complexity, higher costs, and longer acquisition times. Therefore, we compare two image-based methods with the quality assessment based on height data. The first method uses feature vectors of coaxial acquired grayscale images. The significant advantage is that a camera is often integrated into the laser system, so no additional hardware is required. In the second approach, we use an AI-based single-view 3D reconstruction method. The height profile is calculated from a camera image and used for further quality assessment. Thus, we combine the advantages of 2D data acquisition with higher accuracy in evaluating 3D data. In this paper, we analyze a dataset of welded hairpins with different defect types and compare the quality assessment using the height data acquired with OCT, the feature vectors from the camera images, and the reconstructed height data.

Keywords Laser welding, hairpin, quality assurance, OCT, stacked dilated U-Net (SDU-Net), 3D reconstruction

1 Introduction

With the substantial increase in automation of industrial production lines, reliable and also automated quality control is essential. Laser

welding processes are a key technology for many industrial applications and must fulfill high-quality requirements [1]. However, various influencing factors can lead to defects in the weld seam, which can impair the quality and functionality of the product and result in safety-relevant defects [2, 3]. Therefore, the companies use strict criteria for welding quality.

An increasingly important application with high-quality requirements for laser welding comes from e-mobility. E-mobility will become more and more prevalent in individual transportation in the future. This is why vehicles' designs and various components are constantly refined and optimized. For the new generation of motors, automotive manufacturers increasingly use stators with so-called hairpin technology. The conventional copper windings in the stator of an electric motor are replaced by thick copper rods that are welded together, which saves space and improves the efficiency of an electric motor. Depending on the motor design, between 160 and 220 pairs of copper bars are inserted into the sheet metal stacks of a stator, and the ends are connected, usually by laser welding [4–6]. To ensure the high quality of the entire stator, each weld must be checked for a defect [5, 7]. Different properties and measured variables can be used to evaluate the quality of the weld seam [7, 8]. Various works show that the evaluation of three-dimensional data provides higher accuracy than the analysis of two-dimensional camera images [8–10]. The disadvantages are higher hardware costs, system complexity, and longer process times.

This work presents an approach that computes the height map from a camera image instead of acquiring it with a 3D sensor. This procedure allows us to use the height data for quality assessment without the disadvantages mentioned above. We perform the 3D reconstruction algorithm using a convolutional neural network [11]. The rest of this paper is organized as follows: Section 2 discusses the state of research in welding quality evaluation of hairpins and using a 3D reconstruction algorithm. Section 3 describes the experimental setup and investigations of the generated dataset. Building on this, section 4 introduces different approaches for predicting the hairpin quality from image data, 3D data, and reconstructed 3D data. In section 5, the results are discussed before section 6 provides a summary as well as an outlook for future research activities.

2 Related work

There are a variety of quality monitoring and control systems for laser welding. The use of machine learning (ML) methods is evaluated in [12] and [7]. Unlike many ML applications, the amount of data samples in the industrial environment, especially in research, is limited, and the computing time may not extend the production time [13].

In [14] a post-inspection of laser welds is performed based on images using semantic segmentation. Here, a tiny network structure is used for the reasons just mentioned. [7] uses images from 3 perspectives, front, top and back, to evaluate the seam quality of hairpins. More information about the seam connection can be obtained through the different views. However, integration into a production line is more complex because it is often difficult to attach cameras to the side. The resulting accuracy of the network is in the range from 61% to 92% [7]. [8] analyze and compare different Convolutional Neural Networks (CNN) to perform post-process quality control of hairpins. In addition to 2D grayscale images, 3D scans are used as input to the CNN. Based on the 3D scans, the classification accuracy is higher than using the 2D images. This result supports the assumption that the height values contain relevant information for quality assessment. In [15] and [10], a height profile is also used to determine weld quality in laser welding. Especially in hairpin welding, the height difference between the pair of hairpins before and after welding provides information about the volume of the molten material. This volume, together with the other measured parameters of the surface profile of the weld, is crucial for the welding quality of the hairpins [9].

Due to the cost, higher system complexity and acquisition time, it is advantageous to calculate the height profile using a method of 3D reconstruction. [16] use shape from shading (SFS) to perform a 3D reconstruction of a weld seam. Based on the curvature features, the weld quality is evaluated. Especially in the classification task of complex welds with complex structures and characteristics, the curvature feature contains limited information and cannot be applied to this task. The SFS algorithm reconstructs a shape based on shading variation, assuming a single point light source and Lambertian surface reflectance, where the brightness of an image pixel depends on the light source direction and the surface normal. Due to the hairpins' height and the

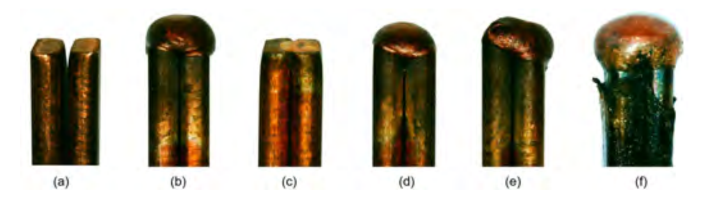


Figure 1: Welding results of hairpins. (a) no weld, (b) good weld, (c) pin not in the focus of the laser, (d) weld with too low power, (e) misaligned pin pair, (f) insulated copper rods.

welding bead's dome, a reconstruction from a single image with SFS is impossible since the incidence of light can only be realized on one side and the other side is accordingly in shadow. [17] calculates a 3D reconstruction from several images taken with different relative positions between camera and weld during the data acquisition phase. Based on the resulting 3D model, a quality evaluation of the weld is performed.

Deep learning-based methods for 3D reconstruction have shown promising results in various research fields. While classical methods deal with shape and image properties such as reflection, albedo, or light distributions, deep learning-based methods use complex network architectures to learn the correlations between 2D and 3D data. Many approaches are challenging to integrate into existing industrial processes because new cameras or illumination equipment are required. [11] compare different single-image reconstruction methods on an industrial dataset. In their investigations, a variation of the U-Net, the stacked dilated U-Net (SDU-Net), has prevailed with its performance.

3 Material

Laser-welded pairs of copper pins, as shown in Figure 1, are used for data acquisition. Different welding results are recorded to obtain a representative data set that includes error cases. Data from 953 hairpins were acquired from a position above the pins, as this perspective allows the integration with the existing industrial process. The 2D intensity images of the hairpins were captured using a Baumer VCXG-15M.I industrial camera based on CMOS technology. An optical coherence tomography (OCT) scanner from Lessmüller Lasertechnik is used to

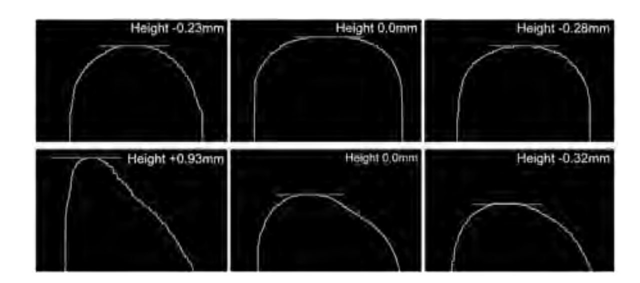


Figure 2: OCT scans at different positions (left side, center, right side). Top row - similar height values indicate a good seam. Bottom row - the different heights indicate the fault case (misaligned pin pair).

capture the 3D data. Many line scans are performed to obtain the height maps of the entire weld. These are then combined to create an overall height map of the component. The exact structure of the data acquisition and the assignment of the camera data to the height data is explained in detail in [11]. To reflect the real situation in the industry with low data availability, we use 10% of the data, i.e., 95 samples, for algorithm development. The other 90%, i.e., 858 samples, are used for testing and evaluation.

4 Detection of weld quality

To compare the result of quality assessments, we analyze various input data for the weld inspection. We use the height data acquired by the OCT, camera images, and reconstructed height data to create feature vectors.

4.1 Height data acquired with OCT

The OCT sensor measures the relative height differences within the weld seam. Good welding of a pin pair results in a round welding bead, which has its maximum in the center. The line scans should have a structure like the upper row in Figure 2 over the entire weld bead. The bottom row shows the images at the same positions of a weld with misaligned pins for comparison. As in [18] and in [19], we compare

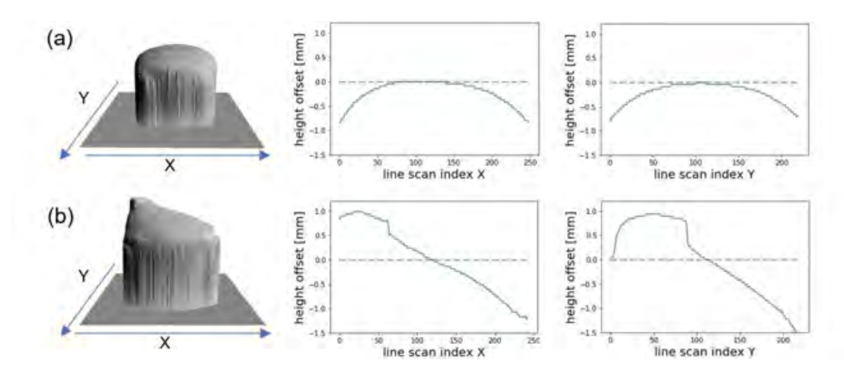


Figure 3: Difference of the maximum values of the line scans to the center. The maximum value of each line scan is determined. The difference to the center is calculated and the values are plotted in a curve. Mathematically this means $f(l_i) = |h_c - \max l_i|$, where l_i is the line scan with index i and h_c is the height value in the center. (a) Good welds result in a curve with its maximum in the center. (b) Defective welds, such as misaligning pins or pins that are not in the laser's focus, can be detected in the curve.

multiple line scans with each other. For higher accuracy, we scan the hairpin in the x- and y-directions with lines at distances of 18 µm.

For quality assessment, we use different criteria. Analogous to [18], we consider the difference between the maximum height values of the individual line scans to the height of the pin center. Through this comparison, we can detect misalignment of the hairpins or misshapen welding beads. The procedure is visualized in Figure 3. In addition to the curve profile, we evaluate the line scans' maximum and minimum distance to the pin center's height. If the distance to the pin center is too small, the weld is not sufficiently stable. If, on the other hand, the minimum distance is too large, this provides information about pores or cracks in the pin surface. We also consider the width of the weld bead in the evaluation.

4.2 Camera images

As mentioned earlier, it is not always possible to capture the height profile due to time constraints and the increasing cost and complexity of the system. Therefore, we develop a different approach by deriving



Figure 4: Detection of the welded and unwelded pin surface in the camera image. The detection of the surface of the weld, as well as the unwelded pins, is shown. In each case, the right image shows the binary mask overlaid on the image (green - weld, red - unwelded pin). (a) good weld, (b) misaligned pin pair, (c) pin not in the focus of the laser, (d) insulated copper rods.

the quality-relevant properties of the weld from the grayscale image. As with OCT scans, we can also infer the width of the weld from the grayscale image. In addition, the size of the weld surface provides information about the stability of the weld. We can also detect this size in 2D images. For the detection of the seam area, threshold-based methods reach their limits due to the low-intensity differences and contrasts in the images. However, CNN-based semantic segmentation can detect the area well, even in small network architectures. Analogous to [6], we train a small SDU-Net to detect both the welded seam and the non-welded pin regions. The predicted masks are shown in an overlay representation in Figure 4.

We can already detect many defect cases by evaluating the width of the weld and the size of the two classified areas. As a further evaluation, we analyze the shape of the weld. In good welds, this is approximately circular and has no solid corners and edges. However, if too little material is melted during welding, no round weld bead is formed, and the contour is slightly angular due to the pin shape. Other defects, such as copper pins that have not had their insulation stripped, also result in edges in the weld shape. Since the weld surface is a closed contour, Fourier descriptors can be used to characterize it. Analogous to [20], we compute the Fourier descriptors of the contours. An evaluation of the harmonics considers the complexity of the contour. In particular, in combination with the information about the size of the non-welded pin region, this contains information about insufficiently welded pin pairs. The relationship between the defined features and the evaluation result of the seam quality based on the height profile is shown in Figure 5.

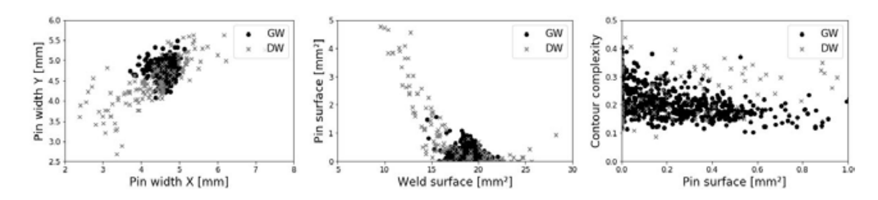


Figure 5: Quality-related features derived from the grayscale images. The correlation of the features derived from the 2D image with the seam quality based on the height profile is shown (GW -good weld, DW - defective weld).

4.3 Height data from the 3D reconstruction algorithm

In the third approach, we use an AI-based single-view reconstruction method. Thus we combine the advantages of the two methods just presented. This approach calculates the height profile from the captured camera image. For this purpose, only one camera image must be taken in the production line, and the algorithm can replace the time-consuming OCT scan. Further analyses can still be performed on the more informative height profile. We use a modified SDU-Net architecture for the reconstruction. Since the model is tiny, with only 162,423 parameters, it can also be executed efficiently on industrial hardware. The exact implementation, the training parameters and the result analysis with deviations from ground truth are explained in detail in [11].

5 Results and discussion

The quality assessment of the 858 test samples is performed separately with each method to evaluate the different approaches. The ground truth is the division into good weld (GW) and defective weld (DW) based on the features derived from the entire recorded height map using OCT. We evaluate the quality assessment based on the criteria visible in the camera image (Cam) and the AI-based 3D reconstruction (3D-R) data. When height data is used for quality assessment, only a few line scans are usually acquired due to time constraints. The scanner made by Lessmueller Lasertechnik has a scan frequency of 70 kHz, so a scan of the entire component takes considerable time. Therefore, we use an approach in which only six OCT scan lines (three in the x-

direction and three in the y-direction) are considered in the evaluation (6L). One scan is in the center of the weld, and the other two are on each side. In another investigation, we only consider the three scans in the x-direction in our evaluation (3L). The feature vectors for the quality assessment are defined based on those of the entire height map. The results are presented in Figure 6 using a confusion matrix.

The AI-based 3D reconstruction using the camera images gives the best results of the four methods compared. 842 of the 858 test samples are classified in the same way as with the ground truth data, even if only the camera image was used as input. The discrepancies are due to borderline cases. As described in detail in [11], the model trained on 95 images has an average deviation of 93.5 µm from the ground truth. Due to the rule-based partitioning into GW and DW, in case of doubt, the deviation from one pixel value may yield a different result. One pixel value corresponds to a deviation of 46.8µm in height and a difference of 18µm in width. The borderline cases are welds where the width or the minimum height of the weld bead was barely reached with one method and just missed with the other.

When evaluating the results based on the camera images, it is noticeable that more pin pairs with height offset were detected as GW. This wrong classification can be attributed to the fact that the height offset is not considered in any of the used image-based classification features. The offset cannot be identified by the shape, size of the weld bead or the area of the unwelded pin surface. Therefore, this error case unfortunately often remains undetected. On the other hand, samples that are incorrectly classified as DW can be attributed to tiny weld beads. If less material was melted during the process, the welds often have a rather rectangular shape due to the pin shape. In some cases, the height of the weld is sufficient to create a stable weld, although it still has an edged shape. Based on the camera image, these samples are classified as DW because they look very similar to the unstable low-power welds. GWs with a round weld bead are reliably detected as GWs.

The evaluation with a few line scans also shows more deviating results than the evaluation with 3D reconstruction. In addition to borderline cases, these methods incorrectly classify pin pairs in which one of the pins was only partially connected or welds with spatter as GW. Especially when evaluating with only three scans in the x-direction, insufficiently welded pins (e.g. Figure 1(c, d)) were missed more often.

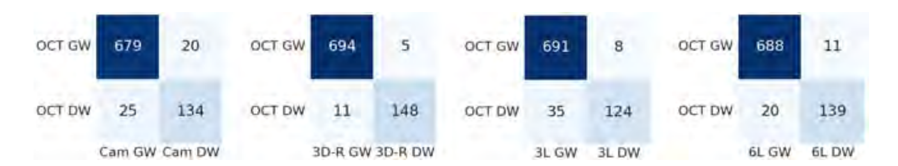


Figure 6: Comparison of the results of the different methods. The results of the approaches: Camera image (Cam), AI-based 3D reconstruction (3D-R), six line scans OCT (6L) and three line scans OCT (3L) are compared with ground truth based on the features from the entire height map.

6 Conclusion

We have developed and compared different methods for quality assessment in hairpin welding. In addition to analyzing the acquired height profile, we have successfully determined the quality based on a grayscale image. For the image-based evaluation, we used two different approaches. First, we used features derived from the image, such as the width and shape of the weld, to perform a quality assessment. The most significant deficiencies were pin pairs, which have an offset between the pins. This misalignment is not captured in the image-based features and, thus, is not considered in the quality assessment. With this approach, the misalignment would have to be checked and corrected before welding, completely avoiding the faulty weld. The significant advantage of using the image-based features is that no additional height scanner is needed, which reduces cost, setup effort, and acquisition time and allows quality analysis through a software update. The calculation of the binary mask following the approach of [6] only requires 16 ms on an i5-7300U CPU. It can be integrated into the process with the subsequent algorithmic evaluation without additional hardware requirements. In a second approach, we performed an AI-based 3D reconstruction on a single grayscale image and then used the computed height data for quality assessment. With this approach, we achieved higher accuracy and could correctly assign the test samples, except for some borderline cases. The approach presented in [11] allows reconstruction based on a single grayscale image. For this purpose, a small SDU-Net architecture is used, which can be executed on an i5-7300U CPU in only 45 ms. This method opens up a new pos-

sibility for quality evaluation. Unlike feature-based evaluation of the camera image, a height scanner is required to train the AI model. Afterward, however, only one camera image is needed in the productive system, and the time for the height scan can be saved.

In future work, we will integrate the developed solutions into the manufacturing process and evaluate the results on other components than hairpins. In addition, the robustness and transferability of an AI model for calculating the height profile between different plants will be further investigated. Depending on the results, it might be necessary to improve the networks or the algorithms used downstream for quality assessment.

References

1. M. Jäger, S. Humbert, and F. A. Hamprecht, "Sputter tracking for the automatic monitoring of industrial laser-welding processes," *EEE Trans. Ind. Electron*, vol. 55, no. 5, pp. 2177–2184, 2008.
2. M. Zhang, G. Chen, Y. Zhou, S. Li, and H. Deng, "Observation of spatter formation mechanisms in high-power fiber laser welding of thick plate," *Appl. Surf. Sci.*, vol. 280, pp. 868–875, 2013.
3. A. Kaplan and J. Powell, "Laser welding: The spatter map," *29th Int. Congr. on Appl. of Lasers and Electro-Optics (ICALEO)*, vol. 103, pp. 683–690, 01 2010.
4. T. Ishigami, Y. Tanaka, and H. Homma, "Development of motor stator with rectangular-wire lap winding and an automatic process for its production," *Electr. Eng. JPN.*, vol. 187, no. 4, pp. 51–59, 2014.
5. T. Glaessel, J. Seefried, and J. Franke, "Challenges in the manufacturing of hairpin windings and application opportunities of infrared lasers for the contacting process," in *7th Int. Elec. Drives Prod. Conf. (EDPC)*, 12 2017, pp. 1–7.
6. J. Hartung, A. Jahn, O. Bocksrocker, and M. Heizmann, "Camera-based in-process quality measurement of hairpin welding," *Appl. Sci.*, vol. 11, no. 21, p. Art.Nr. 10375, 2021.
7. A. Mayr, B. Lutz, M. Weigelt, T. Gläßel, D. Kißkalt, M. Masuch, A. Riedel, and J. Franke, "Evaluation of machine learning for quality monitoring of laser welding using the example of the contacting of hairpin windings," in *8th Int. Elec. Drives Prod. Conf. (EDPC)*, 2018, pp. 1–7.

8. J. Vater, M. Pollach, C. Lenz, D. Winkle, and A. Knoll, “Quality control and fault classification of laser welded hairpins in electrical motors,” in *2020 28th Eur. Signal Process. Conf. (EUSIPCO)*, 2021, pp. 1377–1381.
9. N. Deyneka-Dupriez, “Implementing oct for industrial weld monitoring,” *Laser Syst. Europe*, 09 2019.
10. C. Stadter, M. Schmoeller, M. Zeitler, V. Tueretkan, U. Munzert, and M. F. Zaeh, “Process control and quality assurance in remote laser beam welding by optical coherence tomography,” *J. Laser Appl.*, vol. 31, no. 2, 2019.
11. J. Hartung, P. M. Dold, A. Jahn, and M. Heizmann, “Analysis of AI-based single-view 3D reconstruction methods for an industrial application,” *Sensors*, vol. 22, no. 17, 2022.
12. A. Mayr, M. Weigelt, M. Masuch, M. Meiners, F. Hüttel, and J. Franke, “Application scenarios of artificial intelligence in electric drives production,” *Procedia Manuf.*, vol. 24, pp. 40–47, 2018.
13. M. Weigelt, A. Mayr, J. Seefried, P. Heisler, and J. Franke, “Conceptual design of an intelligent ultrasonic crimping process using machine learning algorithms,” *Procedia Manuf.*, vol. 17, pp. 78–85, 2018.
14. J. Hartung, A. Jahn, M. Stambke, O. Wehner, R. Thieringer, and M. Heizmann, “Camera-based spatter detection in laser welding with a deep learning approach,” in *Forum Bildverarbeitung 2020*. KIT Scientific Publishing, 2020.
15. G. Ye, J. Guo, Z. Sun, C. Li, and S. Zhong, “Weld bead recognition using laser vision with model-based classification,” *Robot. Comput. Integr. Manuf*, vol. 52, pp. 9–16, 2018.
16. Y. Lei, E. Li, T. Long, J. Fan, Y. Mao, Z. Fang, and Z. Liang, “A welding quality detection method for arc welding robot based on 3d reconstruction with sfs algorithm,” *J. Adv. Manuf. Technol.*, vol. 94, pp. 1–12, 01 2018.
17. P. Rodríguez-Gonzálvez, M. Rodríguez-Martín, L. F. Ramos, and D. González-Aguilera, “3d reconstruction methods and quality assessment for visual inspection of welds,” *Autom. Constr.*, vol. 79, pp. 49–58, 2017.
18. Lessmueller, “Hairpin welding,” accessed: 2022-09-01. [Online]. Available: https://lessmueller.de/tasks/harpin-schweissen/?lang=en
19. M. Baader, A. Mayr, T. Raffin, J. Selzam, A. Kühl, and J. Franke, “Potentials of optical coherence tomography for process monitoring in laser welding of hairpin windings,” in *11th Int. Elec. Drives Prod. Conf. (EDPC)*, 2021, pp. 1–10.
20. F. P. Kuhl and C. R. Giardina, “Elliptic fourier features of a closed contour,” *Comput. graph.*, vol. 18, no. 3, pp. 236–258, 1982.

Semantic segmentation with small training datasets: A case study for corrosion detection on the surface of industrial objects

Dennis Haitz, Patrick Hübner, Markus Ulrich, Steven Landgraf, and Boris Jutzi

Karlsruhe Institute of Technology (KIT)
Institute for Photogrammetry and Remote Sensing (IPF)
Englerstr. 7, 76131 Karlsruhe

Abstract In this research, we investigate possibilities to train convolutional neural networks with a small dataset for semantic segmentation, while achieving the best possible model generalization. In particular, we want to segment corrosion on the surface of industrial objects. In order to achieve model generalization, we utilize a selection of established and advanced strategies, i.e. Self-Supervised-Learning. Besides radiometric- and geometric-based data augmentation, we focus on model complexity regarding encoder and decoder, as well as optimal pretraining. Finally, we evaluate the best performing model against a pixel-wise random forest classification. As a result, we achieve an f1-score of 0.79 for the best performing model regarding the segmentation of corrosion.

Keywords Semantic segmentation, classification, machine vision, surface inspection, corrosion detection, quality assurance

1 Introduction

In the field of machine vision (MV), image segmentation techniques are heavily utilized for the surface inspection of industrial objects [1]. Image segmentation leads to image regions that can represent image texture in a geometrically precise manner. Well established segmentation methods like thresholding, clustering or region growing, however, have the disadvantage of lacking semantic information. Newer

deep-learning-based (DL) segmentation methods based on convolutional neural networks (CNN) are capable of adding semantics implicitly within the training process. These methods are often based on fully-convolutional-networks (FCN), which only consist of convolution layers as learnable layers, besides optional batch-normalization. FCNs can be viewed as functions that map an input image to a map of $n \in C$ scores per pixel, where C denotes a set of class labels. By applying an *argmax* function, the most likely class c is chosen for a particular pixel. While DL-based models outperform pre-DL methods on large datasets, the downside of such models is the potential of overfitting due to the large amount of model parameters. In a lot of practical applications, however, no adequate amount of data is available [2]. Among other applications, common MV tasks in the area of surface inspection lack a sufficient amount of data in order to train a DL-based model to generalize well. Recent advances in DL research target the challenges of small training datasets.
This work aims to utilize a selection of these advanced learning strategies as well as established methods in order to approximate the best possible model generalization. Our scenario includes a barrel as it is used for the storage of low radioactive waste (Figure 1(a)), which we from now on refer to as our object. The training set consists of an RGB image I_{Train} of the unwrapped coat of the object (Figure 1(b)), whereas the test set consists of an RGB image I_{Test} of the bottom. Both sets are labeled to separate the image pixels into eight classes. In our previous work [3], we already utilized the coat for training and also testing, though both datasets were from different areas of the coat and therefore disjunct from each other. In this new work, however, I_{Test} is aquired under different illumination conditions, which sets both I_{Train} and I_{Test} even further apart from each other regarding the image characteristic. For our scenario, we exclusively use I_{Train} and no additional image datasets or unlabeled data for training. Merely, we use I_{Train} without labels within a model training at some point in this work. To train a model, I_{Train} is split up into smaller image patches for model input. We employ established and widely used **data augmentation** (Section 3.1) techniques by applying geometric and radiometric image transformations. Another aspect of our work is **encoder pretraining** (Section 3.3). For this purpose, we train the models with randomly initialized model parameters according to some normal distribution and

with ImageNet-pretrained parameters. As a third encoder pretraining strategy, we employ self-supervised-learning (SSL) [4]. For measuring the impact of model complexity, we undertake a **model selection** (Section 3.2). Therefore, we use two encoders with different depth of the same model family: ResNet18 and ResNet50 [5]. The accompanying decoder architectures are U-net [6] and DeepLabv3 [7]. The stated techniques are stages in a training pipeline, where the best performing technique per stage gets chosen. For comparability, we also employ a pre-DL algorithm to evaluate the results of both learning domains against. A **random forest classifier** [8] (RF) (Section 3.4) therefore is applied within the RGB feature space. The result is a pixel-wise classification without further contextual information.



(a) **(b)**

Figure 1: Barrel in the test facility (a). Within the facility, the image data of the coat and bottom is acquired. Image of the unwrapped coat (b), used for training our models.

2 Related work

The automated detection of structural damage such as corrosion on industrial objects based on image data is an active field of research [9]. Specifically, many research efforts are focused on applying end-to-end DL to the task of corrosion detection [10]. This, however, poses the challenge that DL approaches are typically data-hungry, requiring large amounts of training data, while publicly available, labeled datasets for corrosion detection are few and far between [11]. Furthermore, the visual appearance of corrosion is quite specific w.r.t. the respective target materials and shapes and it is still an open research question to what extent the recently published dataset from [12] can be transferred to specific application scenarios such as the coated steel barrels used in

our work. Thus, while other research addresses the question of how to alleviate the effort of creating large-scale training datasets for corrosion detection, e.g. via crowd-sourcing [13] or efficient labeling tools [14], we focus on how to efficiently use small amounts of training data in a DL context by evaluating the impact of pretraining methods from the fields of SSL in relation to the results of state-of-the-art DL networks on a common small-scale dataset.

DL-based corrosion detection can be approached as a classification problem, where image regions are classified w.r.t. the presence of corrosion in a sliding window manner [15] . Sometimes, the results of a sliding window classification are further post-processed to yield pixel-wise segmentation results, e.g. via the activation maps of patches that have been classified as containing corrosion [16]. Other works aim at detecting corrosion by means of DL-based object detection networks such as R-CNNs [17]. Here, first, instance-wise bounding boxes are regressed which are subsequently refined to pixel-wise segmentation masks. Lastly, as is the case in our work, DL-based corrosion detection can be approached as a semantic segmentation task. In [18], different fully convolutional segmentation networks are comparatively evaluated for the task of segmenting corrosion spots on steel structures. In [19], fully convolutional segmentation is compared against an approach based on R-CNN. As the results are found to not be precise enough, they are refined by a contour-aware postprocessing approach. Lastly, [20] apply DeepLabv3 in a multi-temporal setting for damage-progress monitoring.

3 Methodology

In this section, the methods and the utilized datasets are described. Two methodological strings are applied: One string represents the model pretraining with the application of all possible encoder-decoder-combinations. Aside from that, these models are trained with the baseline dataset, as well as the augmented dataset. The last string is a pixel-wise RF classification within the RGB feature space.

3.1 Data augmentation

As mentioned previously, the baseline dataset for training consists of 102 image patches of size 512×512 px. In the data augmentation process, geometric and radiometric transformations are applied to these patches. The geometric transformations consist of rotations, as well as a combined crop and resize operation. Because CNNs are rotation invariant to only some degree, the distinction between an image patch and its rotated variant should have a positive effect on the generalization capability. The second geometric transformation is a combined crop and resize operation. A crop of an image is chosen randomly and then resized to the original image patch size. The resizing operation utilizes a bilinear interpolation. With this combination, we aim at creating new appearences of texture, which differ from the original image patch. Finally, the radiometric transformation consists of a color space transformation to HSI, where saturation and intensity are randomly varied. The image patch then gets transformed back to RGB. This strategy is applied to simulate different illumination situations.

3.2 Model selection

The model complexity is one aspect of our investigation. Usually in ML, in order to prevent overfitting, one strategy is to reduce the model complexity, or to be more specific, the number of model parameters. In the case of DL-based models, one possibility to achieve this is to consider different depths of a model. Another aspect is the selection of a decoder, which is responsible for upsampling the learned features to a map of classification scores with the size of the original image.

Encoders. We utilize the ResNet architecture [5] for our investigations. This architecture is found quite often in literature as a standard model. ResNets are used with different depths. We employ a ResNet18 as the *small* encoder with a rather low complexity. The ResNet50 on the other hand is selected as the *large* encoder, as it contains 50 convolution layers. Large encoders have the advantage of learning more distinct features in the lower convolution layers, but have more parameters to optimize as a disadvantage regarding small training sets. Because the surface textures of our object are not very complex, we aim for better generalization while not requiring such distinct features by applying

the ResNet18 encoder.
Decoders. For similar reasons as mentioned before, we select the U-net and DeepLabv3 architectures as the decoders, as these are commonly used in the domain of semantic segmentation. The U-net has a feature preservation aspect to it, because of the so-called skip-connections. These skip-connections map the output of a convolution layer to its corresponding transpose-convolution layer on the decoding side. The DeepLabv3 achitecture applies so-called atrous convolutions and atrous spatial pyramid pooling. The former is applied to yield a more dense feature representation in the upscaling process. The latter is applied to include scale invariance to some extent.

3.3 Encoder pretraining

To pretrain the encoder, we apply three different methods: random initialization according to a normal distribution, pretraining on the ImageNet dataset and SSL. For the latter method, this is achieved by training an encoder model within an SSL model and then by applying transfer learning, in order to embed the pretrained encoder into the segmentation architecture, which is done by extracting the encoder from the SSL model and append a decoder afterwards.
The **random initialization** often is the default in popular frameworks in contrast to setting the parameters to some constant value. In our case, the parameters are initialized according to the normal distribution parameterization described in [5], with $\mathcal{N}(0, \frac{2}{n_l})$, where n is derived from the number of input features as well as the filter size and l as a layer index.
ImageNet pretraining is popular, because of the transfer learning aspect. Only the features on the first layers of training are of interest because there, low-level features like edges, point-like shapes or corners are already learned. This can help for faster convergence or maybe even convergence at all. Of course, pretraining on other datasets is also possible, especially if they are semantically related to the follow-up training domain.
The field of **self-supervised-learning** is densely connected to the DL-field with a highly active ongoing research. An SSL model is trained on exclusively unlabeled data. In our case, a contrastive SSL method is applied: SimCLR [21]. This method takes a sample out of the dataset

as a positive sample, and another disjunct sample as the negative sample. For higher distinctiveness, also data augmentation strategies are applied to the positive sample. A contrastive loss is then calculated from both samples for backpropagation. The purpose of SSL methods is to learn feature representations without knowledge about semantics. This is called the pretext task. From there, the underlying model can be extracted and added to a so-called downstream task. This procedure can be viewed as a transfer learning. The downstream task in our case is the semantic segmentation, where the SSL pretrained encoder is embedded into.

3.4 Random forest classification

The application of an RF classifier in the RGB feature space is done for evaluation. DL methods outperform pre-DL learning techniques on benchmark datasets in the most cases. In our use-case with a comparatively low amount of data, however, such methods might still outperform DL models.

4 Experiments

This section describes our experimental setup in the domain of methodology. Our goal is to detect corrosion as segmented image regions. The other classes are rejection classes and therefore not of further interest. We have four classes in total: lacquer, dirt, spots and corrosion. In our investigations we found that an over-classification leads to a better separability between corrosion and non-corrosion.
As mentioned in the Section 1, we use the image of the unwrapped barrel coat I_{Train} as our training dataset. It is an RGB image of size 3072×8763 px. For the specification of image size we use the notation of $height \times width$ throughout this work. The models are trained with smaller image patches with no overlap to neighboring patches, cut from I_{Train}. Those image patches are of size 512×512 px. With this size, we want to preserve as much information of the surface texture as possible through keeping spatial coherence. Splitting I_{Train} into patches results in 102 image patches as a baseline training dataset, which is used for training in the setting of no **data augmentation**. By applying data aug-

mentation, the dataset grows to 8160 image patches.
For the first variant of **encoder pretraining**, the encoder is not pretrained, but initialized randomly. We keep the default procedure of PyTorch, which distributes the model parameters according to the so-called Kaiming initialization [5]. For ImageNet-based encoder pretraining, we download the pretrained model parameters from torchvision. The SSL pretraining is done using a batch-size of 512 for both encoders. The dataset in both cases is the unlabeled baseline dataset. It is trained for 8000 epochs. It should be noted that both the ResNet 18 and ResNet50 are randomly initialized for the SSL pretraining.
The **DL model training** is organized using different combinations with and without data augmentation, with ResNet18 and ResNet50 and with three different pretraining settings for the encoder, namely randomly initialized, ImageNet-, and SSL pretrained. At last, the number of the previously mentioned combinations is doubled be employing a U-net and DeepLabv3 decoder architecture. In sum, 24 models are trained and evaluated.
For **random forest training**, we applied 100 trees with a maximum depth of 8. The dataset used for this training is the baseline dataset with no augmentations. The reason is that the number of datapoints (pixels) is sufficient for pre-DL models to generalize well on the one hand, but also on the other hand, RFs are fairly robust against overfitting in general.
The **evaluation** uses the classification metrics precision, recall, f1-score and overall accuracy. These metrics show the performance for pixel-wise classification for each of the methods in order to make them comparable. Another metric for measuring the global per-class overlap of correct classified regions is the Intersection Over Union (IoU). Further on, the mean IoU (mIoU) as another metric is calculated by averaging all class-specific IoU values.

5 Results

In this section the results of our experiments are shown. Qualitative results in the form of visualizations are depicted in Figure 2(a) to 2(f). For quantitative results Table 1(a) shows the global metrics of all trained models. Table 1(b) shows the best performing DL model and the RF

model with class-wise metrics each.

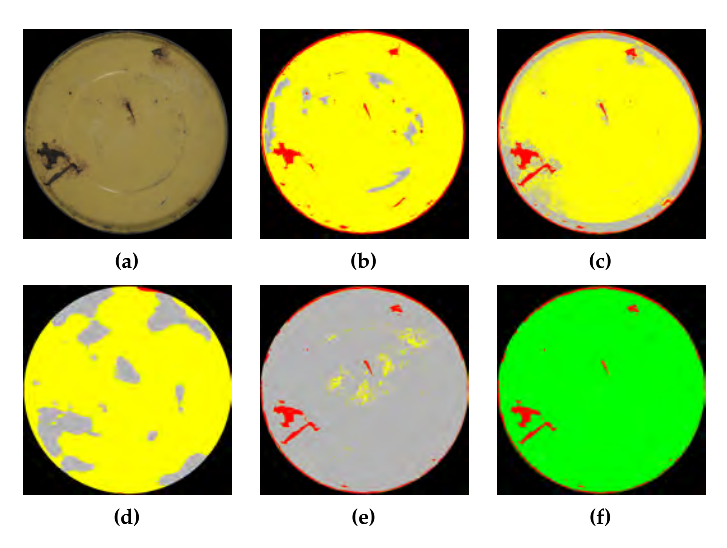


Figure 2: I_{Test} (a), ground truth (b), result of the RF classification (c). Ground truth and predictions are colored as: lacquer (yellow), dirt (bright gray), spots (dark gray), corrosion (red). Predictions of the DL models: worst performing model (*rn50-dl-noaug-inet*) (d), best performing model with all classes (*rn50-u-noaug-rand*) (e), best performing model with two aggregated classes no corrosion (green) and corrosion (red).

6 Discussion

Regarding the **model complexity**, ResNet18 and ResNet50 yield comparable results for the global metrics. For the detection of corrosion, however, ResNet50 usually shows better results. This holds true for the four best performing models concerning the f1-score of the corrosion class. This indicates that lower model complexity does not necessarily lead to better model generalization as proposed in Section 3.
DeepLabv3 and U-net decoders seem to be on par regarding global metrics, as well as for the corrosion detection. The highest f1-score for the corrosion class is achieved by a U-net model. Further on, DeepLabv3 seem to yield more smoothed results in the visual domain,

Table 1: The **global metrics** (a) of each model are shown. The model names are encoded as follows: *encoder-decoder-augmentation-pretraining*. The metrics are mean f1-Score (mF1), overall accuracy (OA) and mean Intersection over Union (mIoU). The marked model is not the best performing regarding the global metrics, but the best performing for the corrosion class. The **class-wise metrics** (b) are shown for the best performing DL model regarding the f1-score of the corrosion class, and the RF model. In addition to f1-score (F1) and intersection over union (IoU), precision (P) and recall (R) are depicted.

(a)

Model	OA	mF1	mIoU
rn18-u-noaug-rand	0.52	0.29	0.35
rn18-u-noaug-inet	0.65	0.37	0.48
rn18-u-noaug-ssl	0.75	0.41	0.60
rn18-dl-noaug-rand	0.81	0.30	0.68
rn18-dl-noaug-inet	0.39	0.20	0.24
rn18-dl-noaug-ssl	0.63	0.27	0.46
rn18-u-aug-rand	0.86	0.36	0.76
rn18-u-aug-inet	0.91	0.39	0.83
rn18-u-aug-ssl	0.82	0.29	0.70
rn18-dl-aug-rand	0.88	0.37	0.79
rn18-dl-aug-inet	0.89	0.39	0.80
rn18-dl-aug-ssl	0.88	0.43	0.78
rn50-u-noaug-rand	**0.10**	**0.21**	**0.05**
rn50-u-noaug-inet	0.80	0.41	0.67
rn50-u-noaug-ssl	0.54	0.38	0.37
rn50-dl-noaug-rand	0.39	0.24	0.24
rn50-dl-noaug-inet	0.75	0.24	0.61
rn50-dl-noaug-ssl	0.63	0.27	0.46
rn50-u-aug-rand	0.86	0.36	0.76
rn50-u-aug-inet	0.91	0.39	0.83
rn50-u-aug-ssl	0.82	0.29	0.70
rn50-dl-aug-rand	0.92	0.42	0.85
rn50-dl-aug-inet	0.86	0.40	0.76
rn50-dl-aug-ssl	0.88	0.40	0.79
random forest	0.80	0.44	0.67

(b)

Model	Class	P	R	F1	IoU
rn50-u-noaug-rand	Laquer	0.86	0.03	0.05	0.10
	Dirt	0.05	0.99	0.09	0.04
	Spots	0.00	0.00	0.00	0.00
	Corrosion	0.83	0.63	0.71	0.21
random forest	Laquer	0.95	0.83	0.86	0.82
	Dirt	0.06	0.21	0.09	0.05
	Spots	0.00	0.00	0.00	0.00
	Corrosion	0.82	0.76	0.79	0.42
rn50-u-noaug-rand	No Corrosion	0.98	0.99	0.99	0.90
	Corrosion	0.83	0.63	0.71	0.21
random forest	No Corrosion	0.99	0.99	0.99	0.96
	Corrosion	0.82	0.76	0.79	0.42

whereas some U-net-based models tend to show slightly more scattered segmentation results.

A surprising insight is that **data augmentation** did not seem to have a positive effect for all models. Moreover, we could only observe in the three best models, regarding f1-score in the corrosion class, that DeepLabv3 decoders benefit from data augmentation and tend to perform poor without data augmentation, while this tends to be the opposite case with U-nets.

For the **encoder pretraining**, we could not observe tendencies re-

garding the different pretraining strategies resulting in a superior performance. This is especially of interest, because random initialization is usually considered as an inferior starting point for training. In our experiments, the random initialization performs similar w.r.t. the other pretrainings. In literature, usually thousands of unlabeled images are utilized for SSL. As can be seen in Table 1(a), no gain could be achieved with SSL pretraining. It can be assumed that the 102 image patches were too few for a substantial SSL pretraining.
The **random forest classification** yields the best results regarding the f1-score of the corrosion class. It needs to be considered, however, that the RF classifier does not take context into account in our experiments. This leads to results with less smoothness in some regions where the separability in RGB space is not very pronounced. Especially larger areas of corrosion are prone to false negatives in the form of scattered pixels belonging to other classes.

7 Conclusion

For our applied strategies in order to train a DL model to generalize from a small baseline dataset, we found that for the core class of corrosion, a RF classifier performs better within the RGB feature space than a DL-based model. The RGB feature space in our case is well separable: There is no surface texture with a similar radiometric signature to that of corrosion in I_{Test}. Also, for the incorporation of context in the non-DL domain, a conditional random field could be of advantage. For the enrichment of the feature space, textural features can be extracted and added for training.
For the DL domain we found that there is still a large potential for improvement. While strategies like data augmentation are mandatory for a long time in such scenarios, we could not see a significant advantage. We only touched the surface of what is possible, with mediocre results at this point. Other possibilities are to incorporate unlabeled datasets for Semi-Supervised-Learning or a large scale Self-Supervised-Learning for better encoder pretraining. Also, Few-Shot-Semantic-Segmentation techniques can be taken into account in the future, as there is a fairly high research activity in this area.

References

1. C. Steger, M. Ulrich, and C. Wiedemann, *Machine Vision Algorithms and Applications*, 2nd ed. Wiley-VCH Verlag, 2018.
2. M. Heizmann, A. Braun, M. Glitzner, M. Günther, G. Hasna, C. Klüver, J. Krooß, E. Marquardt, M. Overdick, and M. Ulrich, "Implementing machine learning: chances and challenges," *Automatisierungstechnik*, vol. 70, no. 1, pp. 90–101, 2022.
3. D. Haitz, B. Jutzi, P. Hübner, and M. Ulrich, "Corrosion Detection for Industrial Objects: From Multi-Sensor System to 5D Feature Space," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B1-2022, pp. 143–150, 2022.
4. L. Jing and Y. Tian, "Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4037–4058, 2021.
5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
6. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
7. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834 – 848, 2017.
8. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
9. S. K. Ahuja and M. K. Shukla, "A Survey of Computer Vision Based Corrosion Detection Approaches," in *International Conference on Information and Communication Technology for Intelligent Systems*, 2017, pp. 55–63.
10. W. Nash, T. Drummond, and N. Birbilis, "A Review of Deep Learning in the Study of Materials Degradation," *npj Materials Degradation*, vol. 37, pp. 1–12, 2018.
11. E. Bianchi and M. Hebdon, "Visual structural inspection datasets," *Automation in Construction*, vol. 139, pp. 1–18, 2022.
12. B. Yin, N. Josselyn, T. Considine, J. Kelley, B. Rinderspacher, R. Jensen, J. Synder, Z. Zhang, and E. Rundensteiner, "Corrosion Image Data Set for

Automating Scientific Assessment of Materials," in *British Machine Vision Conference (BMVC)*, 2021, pp. 1–15.

13. W. T. Nash, C. J. Powell, T. Drummond, and N. Birbilis, "Automated Corrosion Detection Using Crowdsourced Training for Deep Learning," *Corrosion*, vol. 76, no. 2, pp. 135–141, 2020.
14. A. Rahman, Z. Y. Wu, and R. Kalfarisi, "Semantic Deep Learning Integrated with RGB Feature-Based Rule Optimization for Facility Surface Corrosion Detection and Evaluation," *Journal of Computing in Civil Engineering*, vol. 35, no. 6, pp. 04 021 018:1–15, 2021.
15. T. Papamarkou, H. Guy, B. Kroenck, J. Miller, P. Robinette, D. Schultz, J. Hinkle, L. Pullum, C. Schuman, J. Renshaw, and S. Chatzidakis, "Automated Detection of Corrosion in Used Nuclear Fuel Dry Storage Canisters Using Residual Neural Networks," *Nuclear Engineering and Technology*, vol. 53, no. 2, pp. 657–665, 2021.
16. B. Burton, W. T. Nash, and N. Birbilis, "RustSEG – Automated Segmentation of Corrosion Using Deep Learning," *arXiv*, pp. 1–28, 2205.05426.
17. S. K. Fondevik, A. Stahl, A. A. Transeth, and O. Øystein Knudsen, "Image Segmentation of Corrosion Damages in Industrial Inspections," in *IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020, pp. 787–792.
18. L. D. Duy, N. T. Anh, N. T. Son, N. V. Tung, N. B. Duong, and M. H. R. Khan, "Deep Learning in Semantic Segmentation of Rust in Images," in *ICSCA 2020: Proceedings of the 2020 9th International Conference on Software and Computer Applications*, 2020, pp. 129–132.
19. I. Katsamenis, E. Protopapadakis, A. Doulamis, N. Doulamis, and A. Voulodimos, "Pixel-Level Corrosion Detection on Metal Constructions by Fusion of Deep Learning Semantic and Contour Segmentation," in *International Symposium on Visual Computing (ISVC)*, 2020, pp. 160–169.
20. E. L. Bianchi, N. Sakib, C. Woolsey, and M. Hebdon, "Bridge Inspection Component Registration for Damage Evolution," *Structural Health Monitoring*, pp. 1–24, 2022.
21. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

Innovative Qualitätssicherung mittels optimierter Bildverarbeitungsketten auf Basis von Deep Learning

Innovative quality assurance using optimized image processing chain based on deep learning

Katharina Anding[1], Galina Polte[1], Lilli Steinert[1], Daniel Garten[2], Marco Kraft[3], Martin Welzenbach[4] und Claudia Gärtner[3]

[1] Technische Unversität Ilmenau, Fakultät für Maschinenbau, Fachgebiet Qualitätssicherung und Industrielle Bildverarbeitung, Gustav-Kirchhoff-Platz 2, 98693 Ilmenau
[2] GFE Schmalkalden e.V., Näherstiller Straße 10, 98574 Schmalkalden
[3] microfluidic ChipShop GmbH, Stockholmer Str. 20, 07747 Jena
[4] Ziemann & Urban GmbH, Prüf- u. Automatisierungstechnik, Am Bleichbach 28, 85452 Moosinning

Zusammenfassung In diesem Beitrag werden intelligente Qualitätssicherungslösungen für die automatische Erkennung verschiedener Fehlerklassen im industriellen Fertigungsprozess unter Optimierung der Bildverarbeitungs- und Mustererkennungskette auf Basis von Deep Learning diskutiert. Exemplarisch werden intelligente Qualitätssicherungslösungen für die industriellen Fertigungsprozesse Kunststoffspritzguss von mikrofluidischen Bauteilen in der Medizintechnik sowie von Makrobauteilen im Automobilbau aufgezeigt. Die Anwendung leistungsfähiger Deep-Learning-Algorithmen mit ihrem Prinzipbedingt gegebenen höheren Generalisierungs- und Abstraktionsvermögen ermöglicht smarte intelligente In-Prozess-Lösungen zur Evaluierung der Fertigungsqualität und ermöglicht auch Rückschlüsse zum Fertigungsprozess selbst. In diesem Beitrag werden die relevanten Aspekte zur Lösung verschiedener industrieller Qualitätssicherungsaufgaben mittels tiefer neuronaler Netze näher beleuchtet.

Schlüsselwörter Deep Learning, Convolutional Neural Network (CNN), Künstliche Intelligenz

Abstract This paper discusses intelligent quality assurance solutions for the automatic detection of different defect classes in industrial manufacturing processes by optimizing the image processing and pattern recognition chain based on Deep Learning. Exemplary intelligent quality assurance solutions for the industrial manufacturing processes plastic injection molding of microfluidic components in medical technology as well as macro components in automotive manufacturing are shown. The application of powerful deep learning algorithms with their principle-based higher generalization and abstraction capability enables smart intelligent in-process solutions for the evaluation of manufacturing quality and also allows conclusions to be drawn about the manufacturing process itself. In this paper, the relevant aspects for solving various industrial quality assurance tasks using deep neural networks are examined in more detail.

Keywords Deep learning, convolutional neural network (CNN), artificial intelligence

1 Motivation und Ziele der vorgestellten Forschung

Qualitätssicherungsaufgaben im heutigen Produktionsumfeld haben in aller Regel, völlig unabhängig vom Produktionsprozess selbst, die Gemeinsamkeit, dass die automatisierte Qualitätsevaluierung in Form der Produktanalyse nur durch eine Übertragung des Experten-Apriori-Wissens auf ein maschinelles System umgesetzt werden kann. Hierfür werden neben einem Problem-angepassten Bildverarbeitungssystem im Falle der optischen Signalerfassung und –verarbeitung auch eine intelligente algorithmische Umsetzung der Bildverarbeitungs- und Mustererkennungskette notwendig sowie die Zusammenstellung des Experten-Apriori-Wissens in Form von manuell klassifizierten Datensätzen für ein anschließendes Klassifikatortraining. Damit wird deutlich, dass Methoden der Künstlichen Intelligenz (KI) zur Lösung heutiger Qualitätssicherungsaufgaben in der intelligenten, ressourcenschonenden industriellen Produktion unerlässlich sind. Da im Produktionsprozess in aller Regel die existierenden Fehlerklassen

bereits im Vorfeld bekannt sind und die Erkennungsperformance auf einem hohen Niveau liegen muss, finden für derartige Qualitätssicherungsaufgaben überwachte maschinellen Lernverfahren (supervised machine learning) Anwendung. Im Bereich des überwachten maschinellen Lernens gibt es eine Vielzahl von KI-Methoden, sowohl konventionelle als auch Deep-Learning-Methoden. Insbesondere das Deep Learning hat aufgrund vielversprechender Ergebnisse in vielen Bereichen von Wissenschaft, Industrie und Alltagsleben mittlerweile stark an Bedeutung gewonnen. Der Erfolg tiefer neuronaler Netze hängt unmittelbar mit der gestiegenen Rechenperformance und insbesondere der Rechenleistungszunahme der High-Performance-Graphik-Karten zusammen, welche eine Berechnung neuronaler Netze solcher Kapazität überhaupt erst möglich machen.
Mit Deep-Learning-Netzen können grundsätzlich sehr gute Erkennungsraten erzielt werden, wenn entweder vortrainierte neuronale Netze verwendet werden, welche auf der Basis von Bildern ähnlicher industrieller Erkennungsaufgaben vortrainiert wurden, oder wenn sehr große Mengen an vorklassifizierten Trainingsdaten zur Verfügung gestellt werden können (sehr kosten- und zeitintensiv). In diesem Beitrag werden innovative Qualitätssicherungslösungen zur automatischen Erkennung verschiedener Fehlerklassen im industriellen Fertigungsprozess beim Kunststoffspritzguss von mikrofluidischen Bauteilen in der Medizintechnik und Makrobauteilen im Automobilbau untersucht und vorgestellt (siehe Abbildung 1).



Abbildung 1: Prüfteilbeispiele: mikrofluidisches Bauteil aus dem Bereich der Medizintechnik (links) [1] und Makrobauteile aus der Automobilindustrie (rechts) in Form von einem Reflektorbauteil (oben) und einem LED-Gehäusebauteil (unten) [2].

Für beide untersuchte industrielle Qualitätssicherungsaufgaben be-

steht die Notwendigkeit, eine angepasste Bildverarbeitungs- und Mustererkennungskette sowie die Anwendung leistungsstarker KI-Algorithmen mit erhöhten Generalisierungs- und Abstraktionsfähigkeiten zu realisieren. Die Lösungen der untersuchten industriellen Erkennungsaufgaben haben gemeinsam, dass insbesondere innovative vortrainierte Deep-Learning-Netzwerke [3] gute Ergebnisse liefern können. In diesem Beitrag werden die notwendigen Schritte zur Lösung einer automatisierten Qualitätssicherung im Mikrofluidik-Kunststoffspritzguss und im Makrokunststoffspritzguss gegenübergestellt und die verschiedenen Aspekte einer angepassten Bilderfassung und eines Klassifikationsroutinendesigns näher beleuchtet.

Im Kunststoffspritzguss von Makrobauteilen im Automotive wird ein Prüfsystem für die fertigungsintegrierte Prüfung komplex strukturierter Kunststoffbauteile vorgestellt. Das robotergestützte Prüflingshandling ermöglicht sowohl eine vollautomatische Stichproben- als auch eine 100%-Kontrolle und Aussortierung fehlerhafter Bauteile in Abhängigkeit zur gewählten Taktzeit der Spritzgussmaschine. Auf Basis moderner maschineller Lernverfahren wird sowohl die Maßhaltigkeit überprüft als auch die Oberflächenbeschaffenheit auf kleinste Fehler wie Einschlüsse, Blasenbildung, lokale Verformungen oder Farbabweichungen untersucht. Durch den Einsatz adaptiver Prüfmethoden können die Prüfverfahren an neue Bauteilgeometrien, Materialien und Farbmerkmale angepasst werden. Die kurze Prüfzeit ermöglicht einen Echtzeitbetrieb auch bei hohen Durchsatzraten. Damit ist diese Methode im besonderen Maße für den Einsatz im Spritzguss von Makrobauteilen geeignet. Die im Spritzguss hergestellten untersuchten Bauteile kommen in Fahrzeugen zum Einsatz und unterliegen einem hohen Produktionsvolumen bei gleichzeitig kurzen Produktlebenszyklen, was eine schnelle, effiziente, kostengünstige und adaptierbare Lösung notwendig macht. Dank eines speziell entwickelten robotergestützten Bildaufnahmesystems können auch komplexe und transparente Objekte inspiziert werden.

Im Kunststoffspritzguss von mikrofluidischen Bauteilen wurde ein innovatives Inspektionssystem zur fertigungsintegrierten Prüfung von komplex strukturierten mikrofluidischen Kunststoffbauteilen für Lab-on-a-Chip-Anwendungen in der Diagnostik erarbeitet. Diese medizintechnische Anwendung von spritzgegossenen mikroflui-

dischen Bauteilen erfordert eine hochpräzise und 100%ige Kontrolle der Produktion. Die optische Qualitätskontrolle wurde durch die Entwicklung eines QC-Prototyps und einer angepassten KI automatisiert realisiert. Auch hier werden Algorithmen aus dem Bereich des maschinellen Lernens zur Auswertung der Bilddaten eingesetzt. Übergeordnetes Ziel war es, die Qualität und Produktivität der gesamten Wertschöpfungskette zu steigern.

2 Stand der Technik der Qualitätssicherung mittels Bildverarbeitung und KI in der industriellen Produktion

Die digitale Bildverarbeitung spielt eine herausragende Rolle in der Qualitätssicherung von Produktionsprozessen. Neben den klassischen Anwendungen ebnete sie auch den Weg für Lösungen im Bereich der Industrie 4.0 [4], [5]. Bei der Oberflächenprüfung werden in der Praxis häufig manuelle Stichprobenprüfungen durchgeführt, die mit einem hohen Zeitaufwand und subjektiven, prüferabhängigen Ergebnissen verbunden sind. Aufgrund immer schnellerer Produktionsprozesse, fortschreitender echtzeitnaher Anforderungen in der Qualitätsanalyse und erheblicher Fortschritte in der Rechen- und Analysetechnik sind manuelle Oberflächenanalysen nicht mehr zeitgemäß, weshalb auch ein schnell fortschreitender Wechsel zu automatisierten Verfahren beobachtet werden kann [6]. Für unterschiedliche Anwendungsbe-reiche existieren bereits verschiedene Methoden zur Oberflächenanalyse unter Verwendung von Bildverarbeitungsmethoden, z.B. die Detektion von Defekten auf Plattenmaterial, die Inspektion von lackierten Oberflächen und die Bildanalyse zur Defekterkennung auf Wafern [7]. Für die Analyse von Oberflächendefekten auf industriell gefertigten Oberflächen werden meist Verfahren der Texturanalyse eingesetzt [8], [9]. Für die Automatisierung von Oberflächeninspektionsaufgaben sind maschinelle Lernverfahren zur Realisierung der Klassifizierung in in-Ordnung (iO) oder nicht-in-Ordnung (niO) bzw. einzelne Fehlerklassen zwingend erforderlich. Beim maschinellen Lernen erkennt der Algorithmus Muster und Regelmäßigkeiten in den ihm zur Verfügung gestellten Beispielen und wendet diese auf prak-tische Prüfungen an. Während des Lernprozesses wird die Korrelation zwischen Merkma-

len und Klassen der Trainingsobjekte ermittelt, die zur Vorhersage der Klassen von unbekannten Objekten genutzt wird [10]. Neben den klassischen Algorithmen erfährt das sogenannte Deep Learning derzeit einen erheblichen Aufschwung. Das Convolutional Neural Network (CNN) zum Beispiel ist ein tiefes künstliches neuronales Netz, das sich besonders für Bildverarbeitungsaufgaben eignet [11]. Während bei klassischen Verfahren relevante Regionen in Bildern segmentiert und wichtige Merkmale berechnet werden müssen, entfallen diese Zwischenschritte bei CNNs, da sie innerhalb des Algorithmus automatisiert erfolgen.
Deep-Learning-Algorithmen haben in den letzten Jahren bemerkenswerte Ergebnisse erzielt und übertreffen die Fähigkeit traditioneller Methoden, Korrelationen in hochdimensionalen Datensätzen zu finden. Dennoch gibt es einige Nachteile und Einschränkungen bei der Anwendung dieser Algorithmen. Tiefe neuronale Netze benötigen eine extrem große Datenmenge, um eine gute Gene- ralisierungsfähigkeit zu entwickeln und damit gute Ergebnisse zu liefern [12]. Alternativ dazu können vortrainierte CNNs verwendet werden, welche im Idealfall bereits mit großen Bilddatensätzen industriellen Ursprungs vortrainiert wurden.

3 Bilderfassung und Datensatzerstellung für die untersuchten industriellen Qualitätssicherungsaufgaben

In Abbildung 2 sind die für beide industrieelle Applikationen erarbeiteten und für die Untersuchungen verwendeten Bildaufnahmeeinrichtungen dargestellt, links im Bild der roboterassistierte Prüf-stand in der Kunststoffspritzgussfertigungsanlage zur Prüfung von Makrobauteilen im Automotive und rechts der Prüfstand für die Prüfung von mikrofluidischen Chips für die Medizintechnik. Beide Systeme arbeiten auf Basis optischer Sensoren, unterscheiden sich jedoch stark in der technologischen Umsetzung aufgrund sehr unterschiedlicher Anforderungen im Fertigungsprozess und den zu prüfenden Bauteilen.

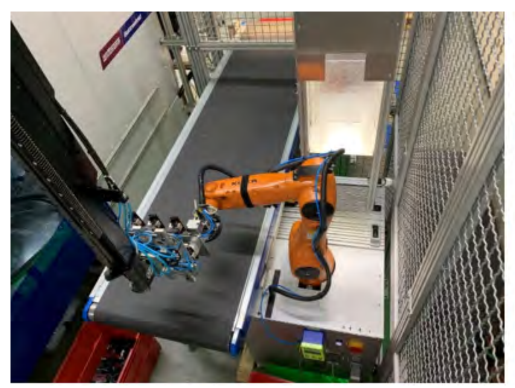



Abbildung 2: Bildaufnahmeeinrichtungen für die Prüfung von Makrobauteilen im Automobilbau (links) [2] und von mikrofluidischen Chips (rechts).

3.1 Bilderfassung und Datensatzerstellung für die Qualitätssicherung von Kunststoff-Spritzgussteilen im Automobilbau

Zunächst wurden mit verschiedenen Kamerasystemen Voruntersuchungen zur Bildaufnahme durchgeführt, um den Materialcharakteristiken und Reflexionseigenschaften der Prüfteile zu entsprechen und qualitativ hochwertige Bilder gewinnen zu können. Die finale Bildaufnahmeeinrichtung besteht aus einer 5-Megapixel-Kamera mit Auflicht und Durchlicht sowie einem Gehäuse zum Schutz vor Fremdlicht. Das Handling der Prüfteile wurde vollautomatisch mittels eines Knickarmroboters realisiert (siehe Abbildung 2 links). Die Kommunikation mit der Spritzgussmaschine wurde über binäre Signale und ein selbst entwickeltes Protokoll realisiert, um eine hohe Stabilität und Sicherheit zu gewährleisten. Das Beleuchtungssystem wurde optimiert, um einen hohen Kontrast bei der Abbildung kleiner Oberflächendetails und Defekte zu erreichen. Schatten, lokale Reflexionen und lokale Über- und Unterbelichtungen wurden durch eine hochdiffuse Beleuchtungscharakteristik ähnlich einer Dombeleuchtung minimiert.
Für die Untersuchung wurden im Projekt zunächst gemeinsam mit den beteiligten Partnern die Prüfteile festgelegt und Kriterien zur Prüfung erarbeitet sowie ein Fehlerkatalog aufgestellt (Fehlerklassen definiert). Hier konnten insbesondere zwei Fehlerarten herausgearbeitet werden:

zum einen fehlerhafte Maschinenparameter (insbesondere fehlender Nachdruck, falsche Temperatur an der Düse, Defekt an der Spritzgußform) und zum anderen eine fehlerhafte Materialzusammensetzung (insbesondere Beimengung von ungeeignetem Kunststoffgranulat, Farbabweichungen). Als Prüfteile wurden aufgrund der besonderen optischen Herausforderungen ein schwarzes LED-Gehäuse und ein transparentes Reflektorbauteil ausgewählt. Im Ergebnis konnten verschiedene Datensätze der Prüfteile „LED-Gehäuse“ und „Reflektor“ gewonnen werden, auf denen die KI-Algorithmen trainiert wurden, um diese später erfolgreich in den Demonstrator integrieren zu können. Der mit dem Roboter-assistierten Bildaufnahmesystem gewonnene Bilddatensatz besteht aus rund 500 Objektbildern für beide Arten von Prüfteilen (roter Reflektor und schwarzes LED-Gehäuse). Die Musterteile wurden zuvor sowohl während der regulären Fertigung als auch im Rahmen einer gezielten Fehlersimulation gesammelt, indem die Prozessparameter und die Materialzusammensetzung so verändert wurden, dass bewusst fehlerhafte Teile unter kontrollierten Bedingungen produziert werden konnten. Für die Trainingsmenge (100 - 150 Beispiele pro Klasse) wurden repräsentative Musterbilder der Hauptfehler beim Spritzgießen in Form von „Düse zu heiß“, „ohne Nachdruck /zu geringer Nachdruck“ und „falsche Granulatzusammensetzung“ sowie fehlerfreie „Gutteile“ (iO-Teile) für das Prüfteil LED-Gehäuse sowie „Gutteile“, „Defekt Einfall“ und „punktförmige Defekte“ für das Prüfteil Reflektor verwendet. Die deutlich erkennbaren Unterschiede zwischen fehlerfreien und fehlerhaften Teilen ermöglichen es, mit einer relativ kleinen Menge von Musterteilen einen repräsentativen Datensatz für das Training der KI zu erhalten.

3.2 Bilderfassung und Datensatzerstellung für die Qualitätssicherung von mikrofluidischen Kunststoff-Spritzgussteilen für die Medizintechnik

Die hierfür erarbeitete Bildaufnahmeeinrichtung besteht aus einem 12K-Zeilenkamerasensor und einem 12M-Pixel-Matrixkamerasensor. Die resultierende Bildgröße beträgt 25000 x 9000 Pixel. Das gesamte Bildaufnahmesystem ist in einem Schutzgehäuse untergebracht, das

vor Fremdlicht schützt (siehe Abbildung 2 rechts). Dies ist von großer Bedeutung, da schon kleine Veränderungen in der Beleuchtung zu einer geringeren Erkennungsrate führen könnten. Das Einsetzen der mikrofluidischen Bauteile erfolgt derzeit noch manuell, während die anschließende Bilderfassung und -auswertung automatisch durchgeführt werden. Für jedes zu prüfende Bauteil werden 26 Bilder mit beiden Kameras aufgenommen. Der klassische Basis-Algorithmus betrachtet und verfolgt vordefinierte Stellen auf dem mikrofluidischen Bauteil, wie µm-große fluidische Kanäle und µm- bis mm-große Hohlräume, und wertet im Anschluss das Bild aus. Im Bereich vorkommender optischer Unregelmäßigkeiten im fluidischen Kanal werden in einem zweiten nachgelagerten Schritt Bildausschnitte der Abmessungen 500 x 500 Pixel, welche die Unregelmäßigkeiten enthalten, ausgeschnitten und zur automatischen Prüfung an die vortrainierte KI übertragen. Der Ursprung dieser Unregelmäßigkeiten kann von unkritischen Lufteinschlüssen um einen Kanal bis hin zu einem kritischen Partikel im Kanal selbst reichen, der die Funktionsfähigkeit des gesamten Bauteils beeinträchtigen kann und daher als funktionskritisch sicher erkannt werden muss. Die Rohbilder, die Vorschaubilder mit Markierungen und alle gesammelten Ergebnisse werden im Ergebnis in einer NAS- und SQL-Datenbank abgelegt und gespeichert.

Die KI-Erkennungsroutine wurde angelernt mit Teilbildausschnitten der Größe von 500 x 500 Pixeln, die aus dem mikrofluidischen Kanal entnommen und von einem menschlichen Experten vorklassifiziert wurden. Der aufgenommene Datensatz besteht aus insgesamt 2.264 Bildausschnitten der vier Klassen: „Kanal_sauber“ (499 Objektausschnitte), „Kanal_mit_Fließlinie“ (500), „Kanal_Grat“ (425) und „Kanal_Partikel“ (840), wobei die Oberklasse „Kanal_Partikel“ wiederrum die drei Partikelklassen: „Partikel_unkritisch“, „Partikel_kritisch“ und „Partikel_unkritisch_Fluse“ enthält. Eine Differenzierung der verschiedenen Partikelklassen gestaltet sich als schwierig aufgrund der sehr unterschiedlichen Lage der Partikel unterhalb, oberhalb oder im Kanal, welche in einer 2D-Bildaufnahme ohne Tiefeninformation nicht sicher detektiert werden kann.

4 Bildverarbeitungs- und Mustererkennungskette

Abbildung 3 zeigt die Bildverarbeitungs- und Mustererkennungskette, die der Lösung jeder Erkennungsaufgabe mit Hilfe des maschinellen Lernens zugrunde liegt und welche die Basis für die aufgezeigten KI-Lösungen bildet.

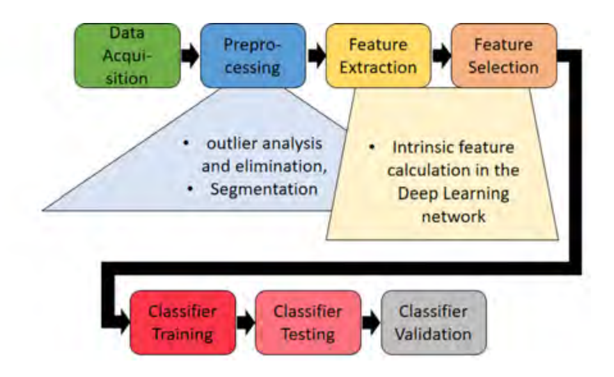


Abbildung 3: Bildverarbeitungs- und Mustererkennungskette.

5 KI-Lösung für die Qualitätssicherung von Kunststoff-Spritzgussteilen in der Automobilindustrie

Ein auf industriellen Bilddaten vortrainiertes CNN der Softwarebibliothek Halcon (enhanced CNN) wurde mit dem gewonnenen Datensatz trainiert und optimiert. Es konnten Erkennungsraten (ER) zwischen 95 und 100 % für die verschiedenen Klassen im Laboreinsatz erreicht werden, wobei reduzierte rgb-Bilder der Größe von 500 x 500 Pixeln verwendet wurden.
Die im Labortest erreichte Erkennungsleistung des besten Deep-Learning-Verfahrens (ebenfalls eine vortrainierte Halcon-CNN) lag für die Prüfteile LED-Gehäuse bzw. Reflektor bei einer mittleren Gesamt-ER von 96,22 % mit einer Standardabweichung (Stabw) von 1,92 % bzw. mittleren Gesamt-ER von 97,63 % mit einer Stabw von 1,70 %.

Die erreichbaren Erkennungsleistungen der einzelnen Fehlerklassen im späteren Robotereinsatz in der Fertigungsstraße, d. h. im Industrieeinsatz unter realen Umgebungsbedingungen, lagen dann zwischen 90 und 100 %. Die im Industrieeinsatz angestrebte und erreichte Klassifikationszeit pro Objekt (ohne Hardware-Handlingzeit) beträgt weniger als 1 ms bei einer mittleren Erkennungsleistung von 90 bis 100 % je nach Einzelklasse. Im Ergebnis konnten damit mit dem finalen Roboter-assistierten System und der auf Industriebildern als Klassifikator vortrainierten CNN der Bildverarbeitungsbibliothek MVTec Halcon (Version 18.11) je nach Kunststoffspritzguss-Applikation (Art des Bauteils), Modellparametereinstellung und vorkommenden Fehlerklassen mittlere Gesamterkennungsraten von deutlich größer 90 % erzielt werden.

6 KI-Lösung für die Qualitätssicherung von Kunststoffspritzguss-Mikrofluidik-Bauteilen in der Medizintechnik

Die für diese Aufgabenstellung verwendete Erkennungsroutine arbeitet mit Teilbildausschnitten der Größe 500 x 500 Pixeln, die vom mikrofluidischen Kanal aufgenommen werden. Das Ziel der Erkennungsroutine war die automatisierte Erkennung von Fehlern des mikrofluidischen Kanals. Teilbildausschnitte wurden der KI zur Bewertung von Fertigungsfehlern zur Verfügung gestellt. Da der Basisalgorithmus die Kanalverfolgung und Bildausschnitterzeugung bereits realisiert, mussten bei dieser Erkennungsaufgabe keine weiteren Vorverarbeitungsschritte durchgeführt werden. Die Expertenvorklassifizierten Teilbilder der iO- sowie der Fehlerklassen (niO) konnten direkt für das Deep-Learning-Netzwerk zur Berechnung der intrinsischen Merkmale und zum Training verwendet werden.
Es wurde eine dreifach stratifizierte Kreuzvalidierung verwendet, d. h. 2/3 der Objekte (Teilbilder) wurden zum Training und zur Validierung und 1/3 zum Testen verwendet. Aus den drei Durchläufen mit iterativ vertauschten Trainings- und Testpartitionen und der über drei Durchläufe gemittelten Erkennungsrate wurde eine statistisch bessere Vorhersagegenauigkeit ermöglicht. Die vier vortrainierten

neuronalen Netze „compact“, „enhanced“, „alexnet“ und „resnet50“ der Softwarebibliothek Halcon 20.05 [3] wurden in ihrer Eignung für das gegebene Erkennungsproblem näher untersucht. Für die verschiedenen CNN-Varianten wurden unterschiedliche, an die Erkennungsaufgabe angepasste Klassifikatorparameter verwendet. Für die Testergebnisse wurden die mittlere Erkennungsrate [%] und die Standardabweichung [%] berechnet. Die durchschnittlich erreichten Gesamterkennungsraten liegen bei 73,99 % für den vortrainierten CNN „alexnet“, 92,92 % für „compact“, 97,17 % für „resnet50“ und 98,32 % für „enhanced“.
Die erreichbaren Erkennungsleistungen der einzelnen Fehlerklassen im späteren Fertigungseinsatz stehen noch aus, da sich der Finaldemonstrator momentan noch im Aufbau befindet.

7 Zusammenfassung der Ergebnisse und Fazit

Dieser Beitrag zeigt die erfolgreiche Anwendung von vortrainierten Deep-Learning-Netzwerken für intelligente Qualitätssicherungslösungen zur automatischen Erkennung verschiedener Fehlerklassen im industriellen Fertigungsprozess exemplarisch für zwei unterschiedliche Applikationen im Kunststoffspritzguss auf. Die vorgestellten KI-Lösungen zeigen eine erfolgreiche Implementierung der Bildverarbeitungs- und Mustererkennungskette und eine herausragende Leistungsfähigkeit vortrainierter Deep-Learning-Netzwerke (CNNs). Die höhere Generalisierungsfähigkeit und das höhere Abstraktionsvermögen von CNNs ermöglichen die Realisierung von vollautomatisierten Qualitätssicherungsprozessen in der industriellen Fertigung. Für beide Anwendungen konnten bei Optimierung der verwendeten KI mittlere Gesamterkennungsraten größer 95 % erreicht werden.

Danksagung

Das vom Freistaat Thüringen geförderte Projekt „OptoCheck - Neuartiges Verfahren zur Inline-Prüfung von Maßhaltigkeit und

Oberflächenbeschaffenheit an komplexen Bauteilen in Maschinenbau und Automotive“ wurde aus Mitteln der Europäischen Union im Rahmen des Europäischen Fonds für regionale Entwicklung (EFRE) kofinanziert. Das Projekt „Digitalisierung und Robotisierung im Kontext von Industrie 4.0 in der Qualitätskontrolle einer mikrofluidischen Detektionsplattform (QualiMikro)“ wurde durch das BMBF gefördert. Besonderer Dank gilt den fördernden Institutionen, die durch ihre finanzielle Unterstützung die Durchführung der Forschung ermöglicht haben. Die Verantwortung für den Forschungsinhalt liegt bei den Autoren.

Literatur

1. “Example for a microfluidic component,” 2021. [Online]. Available: https://www.microfluidic-chipshop.com/
2. D. Garten, M. Ullrich, J. Hilpert, U. Speck, and K. Anding, “Robotergestützte qualitätssicherung in fertigungsprozessen,” in *3. RIS3-Industrieforum, Smarte Fertigung*, January 2021.
3. Halcon, “comprehensive standard software for machine vision with an integrated development environment,” 2022. [Online]. Available: https://www.mvtec.com/de/produkte/halcon/
4. VDMA, *Schlüsseltechnologie für die Automatisierung: Industrielle Bildverarbeitung 2017/18: Anwendungen - Produkte - Bezugsquellen*, 2016.
5. J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. D. Amicis, E. B. Pinto, P. Eisert, J. Döllner, and I. Vallarino, “Visual computing as a key enabling technology for industry 4.0 and industrial internet,” in *IEEE Computer Graphics and Applications, Vol. 35(2), S. 26-40*, 2015.
6. S. Nölling, 2015. [Online]. Available: https://wiki.zimt.uni-siegen.de/fertigungsautomatisierung/index.php/%C3%9Cberblick_zum_Stand_der_Technik_in_der_Oberfl%C3%A4cheninspektion_f%C3%BCr_Metall-_und_Faserverbundwerkstoffe
7. X. Jiang, P. Scott, and D. J. Whitehouse, “Visual computing as a key enabling technology for industry 4.0 and industrial internet,” in *CIRP Annals-Manufacturing Technology, Vol. 57(1), S. 555-558*, 2008.
8. X. Xie, “Visual computing as a key enabling technology for industry 4.0 and industrial internet,” in *ELCVIA Electronic Letters on Computer Vision and Image Analysis, Vol. 7(3), S. 1-22*, 2008.

9. M. Fuß, "Verfahren zur automatisierung der visuellen oberflächeninspektion mit hilfe der bildverarbeitung," 1997, zESS-Forschungsberichte, Dissertation.
10. F. Camastra and A. Vinciarelli, *Machine Learning for Audio, Image and Video Analysis: Theory and Applications*. Springer, 2008.
11. Y. Lecun, Y. Bengio, and H. Geoffrey, "Deep learning," in *Nature, Vol. 521(7553), S. 436*, 2015.
12. G. Marcus, "Deep learning: A critical appraisal," 2018.

Real-time multi-image vignetting and exposure correction for image stitching

Christian Kinzig[1], Guanzhi Feng[1], Miguel Granero[2], and Christoph Stiller[1]

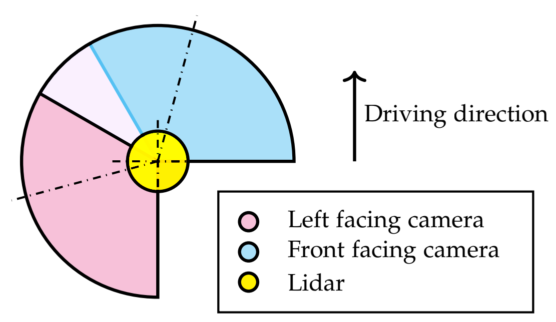
[1] Karlsruhe Institute of Technology (KIT),
Institute of Measurement and Control Systems,
Engler-Bunte-Ring 21, 76131 Karlsruhe, Germany
[2] University of Seville (US),
C. San Fernando 4, 41004 Seville, Spain

Abstract Seamless image stitching depends not only on the accurate alignments of camera images, but also on the compensation of illumination inconsistencies. Even if two images are aligned perfectly, the seam is still visible if the images have a distinct vignetting or different exposure. Image stitching is used to expand the field of view, but a visible seam can lead to significant errors in subsequent visual perception tasks. As a result, we present a straightforward and accurate method for vignetting and exposure correction for stitched images. Firstly, we estimate the camera response function that maps irradiance to intensity. Then, the vignetting model is determined, which is applied to the irradiance images. After that, the exposure of the stitched images is corrected with the irradiance values at the seam. Finally, the irradiance is converted back into intensity using the camera response function. Our approach is evaluated using data recorded by our experimental vehicle and the public nuScenes dataset. Thereby, we test the performance of our method using the IoU of the histograms as well as the mean absolute error of the intensity values in the overlapping image regions. Furthermore, we demonstrate the real-time capability of our approach.

Keywords Autonomous driving, panorama, image stitching, vignetting, exposure, illumination



(a) Sensor setup prototype for UNICAR*agil* project.



(b) Schematic top view of the UNICAR*agil* sensor setup to visualize sensor coverage of color cameras and lidar.

Figure 1: The images from the two lower color cameras of the UNICAR*agil* sensor modul are stitched to a 270° horizontal panoramic image to improve object detection and other perception tasks.

1 Introduction

Autonomous vehicles heavily depend on camera sensors to perceive their surroundings. Object detection, visual localization and mapping are fundamental challenges in automated driving based on camera images. Instead of performing perception tasks for each individual image, the images can be fused to a panorama beforehand [1]. Thus, the horizontal viewing angle can be significantly expanded using image stitching. This facilitates object detection, especially when an object is cut off at the image boundaries by a limited field of view. Image stitching precisely aligns individual images based on image features or lidar measurements. However, the seam is still visible due to vignetting and different exposure times. On the one hand, as shown in [2], vignetting is caused by a radial falloff in irradiance at the image boundaries, while on the other hand, the cameras adjust the exposure time to the current lighting conditions. As a result, the seam between stitched images can lead to false features in object detection and other processing tasks.

In this paper, we propose a straightforward method for compensating vignetting and correcting exposure for multiple stitched images in a time-critical environment. This distinguishes our method from many approaches that aim to compensate for vignetting in individual images using more complex models [3–5]. Hence, we estimate the camera response function (CRF) and the vignetting model before runtime. After the images are stitched, the vignetting model is applied and the exposure is corrected. Our approach is tested on the sensor setup prototype built as part of the publicly funded project UNICAR*agil* [6,7], as well as on the public nuScenes dataset [8]. The prototype of the sensor setup mounted on a vehicle of the Karlsruhe Institute of Technology is shown in Fig. 1(a). In this setup, the camera images from the front-facing camera and the left-facing camera are stitched together to create a panorama. The sensor coverage of the two cameras and the lidar, which allows better alignment of the images, is shown in Fig. 1(b).

2 Related Work

Since cameras are widely used, inexpensive sensors, many articles have already addressed vignetting and exposure correction. Goldman and

Chen provide a good overview of the causes of vignetting and suggest a general approach by modeling the vignetting model as 6th order even polynomial in [2]. In addition, the approaches of Zheng et al. in [3,4] and Cho, Lee, and Lee in [5] focus on vignetting correction for single images. Furthermore, the approach of Kordecki, Palus, and Bal propose the use of a non-radial vignetting model in [9].

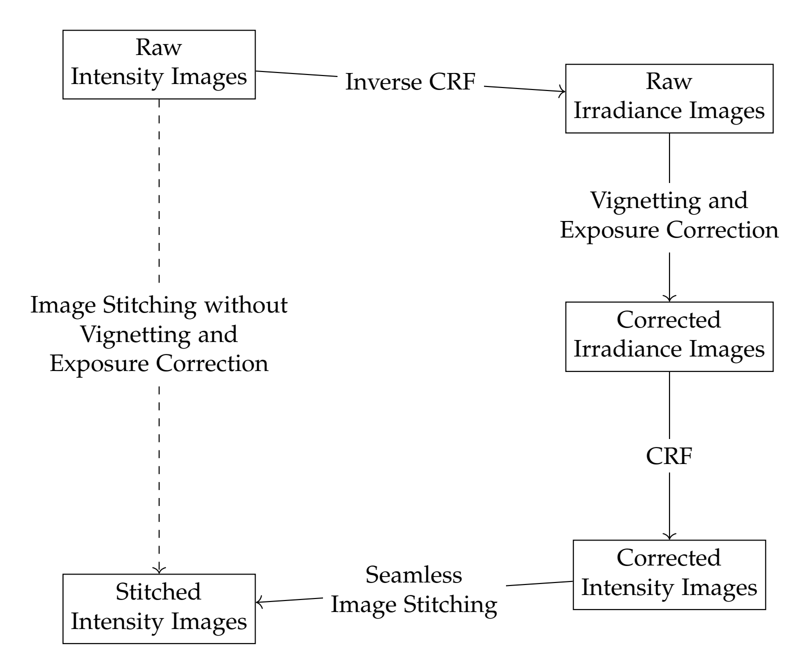


Figure 2: Workflow for vignetting compensation and exposure correction in image stitching.

3 Vignetting and Exposure Correction

Our approach consists of four individual steps. The workflow of our approach is depicted in Fig. 2. First, the camera response functions of all cameras are estimated. Then, the vignetting model is generated and applied. Afterwards, the correction of the exposure between the stitched camera images is performed. In the last step, the corrected



(a) Image stitching without vignetting and exposure correction.



(b) Image stitching with vignetting compensation but without exposure correction.



(c) Image stitching with vignetting and exposure correction.

Figure 3: Comparison of image stitching with and without vignetting compensation and exposure correction. The images are recorded with the UNICAR*agil* sensor setup prototype in Fig. 1.

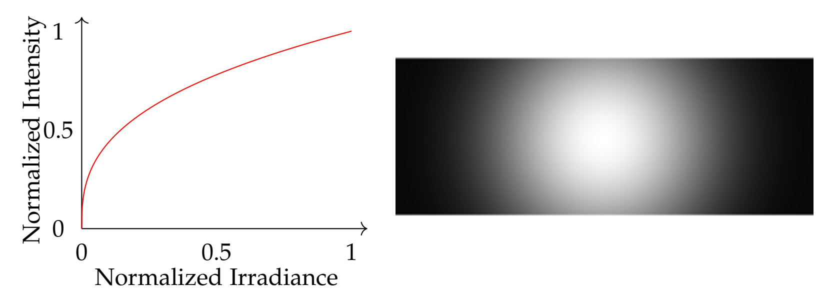
irradiance values are converted back into intensities. The quantitative effect of our approach is shown in Fig. 3 for an exemplary pair of stitched images.

3.1 Calibration of the Camera Response Function

Both the vignetting compensation and the exposure correction are performed based on the irradiance, which is calculated from the intensities using the non-linear camera response function. Therefore, we determine the camera response function of our sensor setup before the actual runtime. The camera response functions are estimated by exposure series in a static scene with known exposure times as in [10]. For each of the color cameras in Fig. 1, we obtain three response functions for the three color channels. However, we found that the camera response functions are approximately identical for the cameras and all color channels. The qualitative evaluation of UNICAR*agil* sensor data shows decent results for vignetting and exposure correction using the approximated camera response function. For this reason, we store only the approximated camera response function for the entire panorama, which is shown in normalized form in Fig. 4(a). After vignetting and exposure correction in 3.2 and 3.3 the camera response function is used to convert the irradiance values back to intensity values.

3.2 Estimation of the Vignetting Model

To compensate for vignetting, we found that in our case a model can be sufficiently created by approximating the vignetting by the cosine-fourth-power law. This estimates the radial irradiance falloff at the boundaries of the camera images. To get a better result for the panoramic image we use a spherical camera model in our approach, that is described in more detail in [1]. As with the pinhole camera model, the intrinsic parameters can be specified in the matrix $\mathbf{A}$ as in Eq. 1, where f denotes the focal length and (u_0, v_0) describes the principal point. Another advantage of the spherical camera model is that the pixel coordinate is proportional to the angle of incidence. This results in Eq. 2, which models the vignette as a function of the distance to the principal point r and the focal length f. In addition, the variables a and b are used to fit the vignetting model. The values are determined



(a) Approximated and normalized camera response function of the UNICAR*agil* sensor setup.

(b) Normalized vignetting model in the spherical camera frame.

Figure 4: Both the camera response function (a) and the vignetting model (b) are computed before the actual runtime, to be applied to the camera images afterwards.

empirically based on a sequence recorded with the UNICAR*agil* sensor module and result in our case in $a = 3.4$ and $b = 0.1$. In Fig. 4(b) the normalized vignetting model in the spherical camera frame is shown. Just like the camera response function, the vignetting model is also created before runtime to achieve real-time capability.

$$\mathbf{A} = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$\begin{gathered} g(u,v) = a\cos^4(r/f) + b, \\ r = \sqrt{(u-u_0)^2 + (v-v_0)^2},\ a,\ b \in \mathbb{R}_0^+ \end{gathered} \tag{2}$$

3.3 Exposure Correction

After the vignetting model is applied, the brightness between the stitched images is not fully adjusted due to a difference in exposure, as can be seen in Fig. 3(b). To perform exposure correction, an individual exposure compensation factor c is determined for each image. For

this purpose, we calculate the quotient c_{rel} from the cumulative irradiances E along the vertical seam in the overlapping region for a pair of stitched images in Eq. 3. Based on the quotient c_{rel}, we derive in the following step the individual exposure correction factors for the single images in Eq. 4 by an additional constraint to ensures that the average of the factors is equal to one. If a panorama consists of more than one seam n, several quotients $c_{rel,n}$ are obtained, from which the individual exposure factors can be calculated. For multiple stitched images with vertical seams n, we use the transitional condition $c_{right,n} = c_{left,n+1}$. In Fig. 3(c) the stitching result after exposure correction is shown.

$$c_{rel} = \sum_{v=0}^{\text{height}} \frac{E_{left}(u(v), v)}{E_{right}(u(v), v)} \tag{3}$$

$$c_{left} = \frac{2}{c_{rel} + 1}, \; c_{right} = \frac{2c_{rel}}{c_{rel} + 1} \tag{4}$$

4 Experimental Results and Evaluation

We evaluate the presented approach on the UNICAR*agil* sensor setup shown in Fig. 1 as well as on the sequences 1 to 10 of the nuScenes dataset [8]. In the latter case, we use the images from the front-facing camera and the front-left-facing camera to create a panorama. Since we do not know what kind of cameras are used in the nuScenes setup and we cannot reconstruct the camera response function from the available data, we assume a linear response function as an approximation. Besides qualitative results in Fig. 3, we show the performance of our method using two different metrics in 4.1 and 4.2. First, the intersection over union of the histograms in the overlapping region of the stitched images is used. The second metric used is the mean absolute error of intensity differences in the overlapping region. Furthermore, we analyze in 4.3 the runtime of the vignetting compensation and exposure correction for stitched images and show its real-time capability.

4.1 Intersection over Union of Histograms

To measure the accuracy of image stitching, we compare the histograms of the two overlapping regions of the single images. This is done before and after vignetting and exposure correction to show the improvement due to our approach. For a better comparison, the images are converted to 8-bit grayscale so that the histogram values are between 0 and 255 with a bin size of 1. The similarity of two histograms H_i can be measured by calculating the intersection over union (IoU). The IoU between two histograms is calculated according to Equation 5. To prevent the resulting panorama from being extremely over- or underexposed, only pixels intensities with values unequal 0 and 255 are considered for evaluation. Table 1 shows the average IoU values of the histograms before and after vignetting and exposure correction on the recording with our UNICAR*agil* sensor setup and on the nuScenes dataset. The increasing IoU using our approach shows that the histograms of the two overlapping regions are better aligned than without our approach.

$$IoU = \frac{\sum_{n=0}^{255} min(H_{left}(n), H_{right}(n))}{\sum_{m=0}^{255} max(H_{left}(m), H_{right}(m))} \tag{5}$$

Table 1: Comparison of the average IoU of the histograms from the overlapping areas of the stitched images. The stitching quality is compared between using only raw images to processed images using our approach for vignetting and exposure correction for the two different image sequences.

	Raw images	Processed images
UNICAR*agil*	46.29 %	55.94 %
nuScenes	38.43 %	46.45 %

4.2 Mean Absolute Error

Since identical histograms can be derived from different images, we additionally evaluate the local similarity between pixel intensities with the mean absolute error. Compared to Zheng et al. we do not measure the difference to a ground truth vignetting function [3]. Instead, we also use the overlapping regions of the single images and calculate

the mean absolute error as another similarity measure to evaluate the image stitching performance. Thereby, we convert the images to 8-bit grayscale and calculate the mean of the absolute differences pixelwise, as shown in Equation 6. Similar to 4.1, we use only pixel pairs with values unequal 0 and 255 for evaluation. The mean absolute error is calculated by dividing by the number of pixel pairs and averaging it by the number of samples in the sequences. The results before and after vignetting and exposure correction are depicted in Table 2 for the sequence recorded with the UNICAR*agil* sensor setup as well as for the nuScenes dataset. The evaluation clearly shows that the mean absolute error decreases if our approach is applied to the images.

$$MAE = \frac{\sum_{u=0}^{width} \sum_{v=0}^{height} |g_{left}(u,v) - g_{right}(u,v)|}{u \cdot v} \tag{6}$$

Table 2: Comparison of the average mean absolute error of the pixel intensities from the overlapping areas of the stitched images. This allows the comparison between using only raw images to processed images using our approach as in 1.

	Raw images	Processed images
UNICAR*agil*	21.58	7.71
nuScenes	37.56	10.99

4.3 Runtime Analysis

In addition to the metrics, which show an improvement in accuracy, we evaluated the real-time capability of our approach. To optimize our approach in terms of its execution time, we run the processing operations directly on the graphics card. This is an option as soon as the entire image processing in an autonomous vehicle is performed on the graphics card since copying data to and from the graphics card takes a lot of time. This offers further advantages, for example, for object detection with machine learning. The improved runtime is an exceptional feature of our simplified approach to vignetting and exposure correction compared to the approaches in [3,5]. In Table 3, we compare the average runtimes of vignetting compensation and exposure correction on

CPU and GPU. On the computer for evaluation, we use *Ubuntu 18.04.6 LTS* as operating system. As CPU an *Intel® Xeon® Prozessor E5-2640 v3* running at 2.6 GHz with 64 GB of RAM and as GPU a *NVIDIA GeForce RTX 2080 Ti* are installed. The table clearly shows that we achieve real-time capability at a frame rate of 10 Hz with an average processing time of 31.36 ms by using the GPU. Further improvements can be expected on the latest generation of *NVIDIA* graphics cards.

Table 3: Comparison of the average runtimes of our approach on vignetting and exposure correction on CPU and GPU.

	CPU	GPU
Runtime in ms	155.35	31.36

5 Conclusion

In this paper we presented a straightforward and effective method on vignetting and exposure correction for multiple camera images and image stitching. Our approach relies on a known camera response function and a previously estimated vignetting model that are applied on the images to be stitched. First, the irradiance is calculated from intensity using the inverse camera response function and our vignetting model is applied. Then, the optimal exposure correction factors for the single images are estimated from the pixels at the seam to improve the quality of the panorama. After vignetting and exposure correction, the intensities are obtained from the modified irradiance values. In summary, the vignetting of the single images is compensated and the transition at the seam of the panorama due to different exposure is corrected. We evaluated our approach by calculating the IoU between the histograms of the overlapping regions of the stitched images before and after vignetting and exposure correction and have clearly demonstrated that the IoU increases significantly after applying our approach. In addition, we have shown that the mean absolute error of the overlapping regions after vignetting and exposure correction also decreases strongly. Both quantitative results confirm the significant improvement in image stitching quality after using our approach. This can lead to higher precision in object detection and other perception tasks. Finally,

we have also shown that our approach can be executed on a graphics card in real-time. To further extend our approach, we plan to integrate joint optimization of exposure correction factors for multiple seams of a full 360° horizontal panoramic image in the future.

Acknowledgment

This research is accomplished within the project UNICAR*agil* (FKZ 16EMO0287). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

References

1. C. Kinzig *et al.*, "Real-time seamless image stitching in autonomous driving," in *25th International Conference on Information Fusion (FUSION)*, 2022, pp. 1–8.
2. D. B. Goldman and J.-H. Chen, "Vignette and exposure calibration and compensation," in *IEEE International Conference on Computer Vision*, 2005, pp. 899–906.
3. Y. Zheng *et al.*, "Single-image vignetting correction using radial gradient symmetry," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
4. ——, "Single-image vignetting correction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, pp. 2243–56, 2009.
5. H. Cho, H. Lee, and S. Lee, "Radial bright channel prior for single image vignetting correction," in *ECCV*, 2014.
6. T. Woopen *et al.*, "UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts," *27th Aachen Colloquium Automobile and Engine Technology*, pp. 663–694, 2018.
7. M. Buchholz *et al.*, "Automation of the UNICARagil vehicles," in *29th Aachen Colloquium Sustainable Mobility*, 2020, pp. 1531–1560.
8. H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.
9. A. Kordecki, H. Palus, and A. Bal, "Practical vignetting correction method for digital camera with measurement of surface luminance distribution," *Signal, Image and Video Processing*, vol. 10, 11 2016.

10. P. E. Debevec and J. Malik, “Recovering high dynamic range radiance maps from photographs,” in *24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997, p. 369–378.

Machine learning-based multiobject tracking for sensor-based sorting

Georg Maier[1], Marcel Reith-Braun[2], Albert Bauer[3], Robin Gruna[1], Florian Pfaff[2], Harald Kruggel-Emden[3], Thomas Längle[1], Uwe D. Hanebeck[2], and Jürgen Beyerer[1,4]

[1] Fraunhofer IOSB, Institute of Optronics, System Technologies and Image Exploitation, Karlsruhe, Germany
[2] Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany
[3] Mechanical Process Engineering and Solids Processing (MVTA), TU Berlin, Germany
[4] Vision and Fusion Laboratory (IES), Karlsruhe Institute of Technology (KIT), Germany

Abstract Sensor-based sorting provides state-of-the-art solutions for sorting of granular materials. Current systems use line-scanning sensors, which yields a single observation of each object only and no information about their movement. Recent works show that using an area-scan camera bears the potential to decrease both the error in characterization and separation. Using a multiobject tracking system, this enables an estimate of the followed paths as well as the parametrization of an individual motion model per object. While previous works focus on physically-motivated motion models, it has been shown that state-of-the-art machine learning methods achieve an increased prediction accuracy. In this paper, we present the development of a neural network-based multiobject tracking system and its integration into a laboratory-scale sorting system. Preliminary results show that the novel system achieves results comparable to a highly optimized Kalman filter-based one. A benefit lies in avoiding tiresome manual tuning of parameters of the motion model, as the novel approach allows learning its parameters by provided examples due to its data-driven nature.

Keywords Sensor-based sorting, machine learning, visual inspection, multiobject tracking

1 Introduction

Sensor-based sorting provides state-of-the-art solutions for sorting of granular materials. This umbrella term describes a family of systems that enable the physical separation of individual objects from a material stream on the basis of information acquired by one or multiple sensors. Among other fields of application, it is considered a key technology for achieving a circular economy. In distinction to mechanical sorting processes such as screening, wind sifting, or float/sink processes, the technology is sometimes also referred to as indirect sorting [1], since particle classification and separation are performed in separate steps. In theory, any number of classes can be recognized for sorting, and separation into multiple fractions is also possible in principle. In industrial applications, however, the task is preferably implemented as a binary sorting task, i. e., sorting into "product" and "residue", since multi-way sorting requires complex mechanical handling.

The functional principle can be summarized as follows. First, the material is fed into the system by means of a conveyor mechanism. Subsequently, the material is transported further via a transport medium. In the course of the transport, sensor-based data acquisition takes place. The data collected is evaluated with the goal to detect and classify individual particles in the material stream. The result of the classification is the basis for the sorting decision, which is executed by means of an actuator. A particular strength of the sorting technology lies in the variety of industrially available sensors that are suitable for use in sensor-based sorting systems. This results in great flexibility with regard to the detectable material properties and thus the sorting criteria to be applied. Due to their suitability for systems with high material throughputs, imaging sensors dominate at this point.

1.1 Motivation

Current systems use line-scanning sensors, which is convenient as the material is perceived during transportation. In case sorting criteria based on color, shape or texture suffice, line-scan cameras in the visible spectrum are used. However, this yields a single observation of each object only and no information about their movement. Due to a delay between localization and separation, assumptions regarding the veloc-

ity need to be made in order to calculate the location and point in time for separation [2,3]. Hence, it is necessary to ensure that all objects are transported at uniform velocities. This is often a complex task.

Recent works show that using an area-scan camera instead of a line-scanning one bears the potential to decrease both the error in characterization [4] and separation [5] in sensor-based sorting. Using a sufficiently high frame-rate, individual objects are observed at multiple time points. By employing a multiobject tracking system, this enables an estimate of the followed paths as well as the parametrization of an individual motion model per object. The latter allows for accurate predictions regarding which actuators need to be activated at what point in time such that an object is deflected and hence removed from the material stream. Therefore, the approach is also referred to as predictive tracking. Eventually, this results in an increased sorting quality.

While previous works focus on physically-motivated motion models, it is shown in [6] that state-of-the-art machine learning methods provide a powerful tool for achieving an increased prediction accuracy, particularly in complex sorting scenarios. However, the approach has not been evaluated in real sorting experiments yet, but rather using pre-recorded image data and a simulated separation.

1.2 Contribution

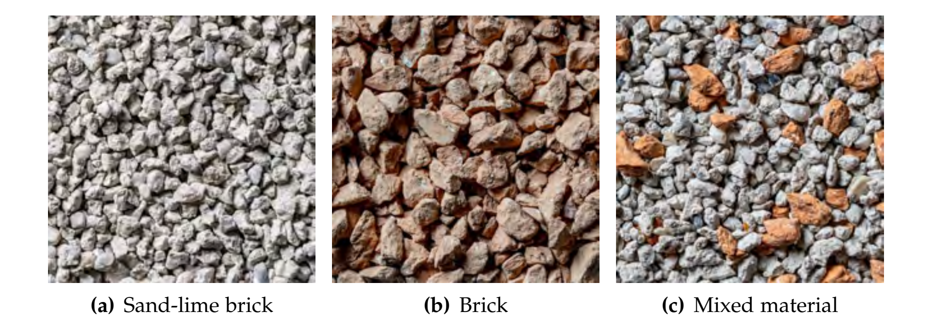
In this paper, we present the development of a neural network-based multiobject tracking system and its integration into a laboratory-scale sorting system with an area-scan camera. This is the first time that the complete development cycle required to make such machine learning-based methods applicable in an industrial sorting setting is considered. With respect to the data processing model itself, we consider the multilayer perceptron from [6]. This model takes observation coordinates of individual objects, which in our case are determined by means of real-time image processing, as an input and generates the predictions for future time points, in our case for the separation stage, as an output. Eventually, actual sorting experiments using the neural network-based multiobject tracking system are conducted.

2 Materials and Methods

In the following, we provide details on the experimental setup, e. g., the exemplary sorting scenario and the considered sorting system, the different prediction models that are compared experimentally as well as the implementation of the real-time inference engine.

2.1 Experimental Setup

We choose an exemplary sorting scenario from the field of construction waste recycling. By generating pure fractions from construction and demolition waste, the material is prepared for the production of recycled construction materials [7]. In our scenario, we consider an input stream consisting of sand-lime brick and brick, see Figure 1. The task is to remove brick from the waste stream. The material is crushed to a grain size of 4 to 6 mm prior to sorting.



(a) Sand-lime brick **(b)** Brick **(c)** Mixed material

Figure 1: Photos of the materials used for the exemplary sorting task.

Both for the acquisition of training data as well as the experimental validation, we use the lab-scale sorting system shown in Figure 2. A detailed description of the system is provided in [5]. A vibrating feeder is used to feed the material in the system. For transportation, a conveyor belt with a width of 140 mm is used. At the end of the belt, right before discharge, the material stream is recorded using an area-scan camera in combination with a ring light. After discharge and during a flight phase, separation is performed using fast switching pneumatic

valves.

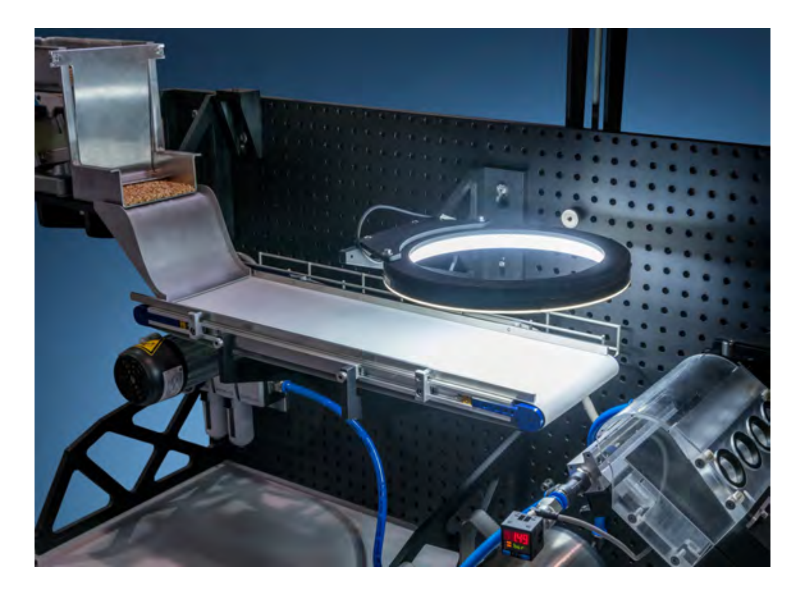


Figure 2: Photo of the lab-scale sorting system used in this study.

The acquired image data is processed with the aim of localizing and classifying individual particles. Based on the classification, a sorting decision is calculated. In case a particle is to be removed from the material stream, a control signal is calculated and transmitted describing the time as well as the valves to be activated in the array. Exactly this calculation, referred to as the prediction model in the following, is the subject of the present study.

2.2 Prediction Models

We validate the proposed approach comparatively. Hence, we also consider two established prediction models for the calculation of the control signals for separation.

First, as a base-line, we consider the system to be equipped with a line-scan camera instead of an area-scanning one. This corresponds to a setup as used in the industry at the time of writing. In this case,

no information regarding particles' motion is known. Consequently, a uniform transport velocity has to be assumed. A fixed, typically experimentally determined delay is added to the point in time of observation of a particle in order to calculate the temporal component of the prediction. Furthermore, it is assumed that no velocity perpendicular to transport direction exists. Hence, the valves to be activated correspond to the lateral position of the particle as seen by the camera.

Second, we consider the approach originally proposed in [8] and experimentally validated in [5]. By using a high-speed area-scan camera, particles contained in the material stream are observed at multiple points in time and tracked via a multiobject tracking system. This way, motion parameters, e. g., the velocity in and perpendicular to transport direction, can be determined individually for each particle. In combination with a motion model, these parameters are used to precisely estimate the control signal for separation. The approach focuses on applying Kalman filters on the centroid of the particles for predictive tracking. In this course, linear, physically motivated models, such as constant velocity (CV), are used.

The novel data-driven approach experimentally validated in this paper takes the last five captured position measurements of each particle as input and directly outputs the control signal for separation, i. e., the estimated arrival time and location of the particle at the separation bar. This is opposed to the original predictive tracking algorithm, which for this purpose uses the estimated positions and velocities from the underlying Kalman filter. The input measurements are provided by the exact same multiobject tracking system employed in the original predictive tracking setup. The approach uses a multilayer perceptron with four hidden layers as a predictor, where each hidden layer consists of 16 neurons. Further details on the architecture and training procedure are given in [6].

While numerous tools and software frameworks are now established for model development, the use of neural networks in production systems and, in the present case, under real-time conditions still represents a very current research topic. In the course of this study, various frameworks for integration into the sorting system were investigated in a first step. A technical constraint was the use of the programming language C++. After a first research, the frameworks *TensorRT* from NVIDIA and *OpenVino* from Intel were chosen. These frameworks dif-

fer fundamentally in the target hardware on which the inference is executed. *TensorRT* allows the execution of the inference on dedicated NVIDIA graphics cards, *OpenVino* on Intel CPUs as well as integrated Intel GPUs. In both cases, conversion of the model was necessary prior to any potential application. *Onnx* was identified as the current supposedly universal format for this purpose.

In addition to training the developed model on the basis of the generated image sequences, it was also necessary to take knowledge about the system structure into account in the implementation, see Figure 3. Here, parameters relating to the separation, such as the distance between the camera observation area and the separation bar, were primarily decisive. To compensate for errors potentially arising due to measurement inaccuracies, parameters for manual configuration of an offset, e. g., with regard to the distance, were implemented.

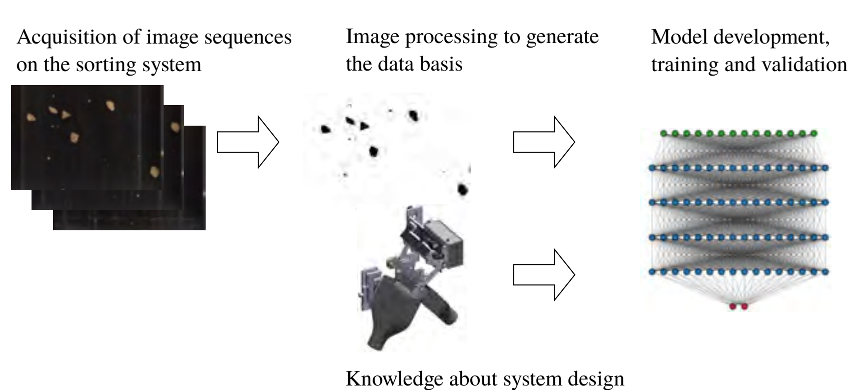


Figure 3: Schematic illustration of the development process of the machine-learning based multiobject tracking.

3 Experimental Validation

We conduct sorting experiments using the methods and materials described in Section 2. One experiment corresponds to sorting 200 g of the material in a batch manner. Additionally to the three prediction models described in Section 2.2, three different mixing ratios are investigated. More precisely, we consider ratios of *residue*, i. e., brick, of

10 %, 25 % and 50 %. Furthermore, we conduct experiments with a mass flow of 10 g/s and 20 g/s.

3.1 Model Training

The multilayer perceptron was trained on a data set of particle tracks recorded on the lab-scale sorting system described in Section 2, with tracks obtained by a preceding offline run of the multiobject tracking algorithm. Although we test the novel approach on several mass flows and mixing ratios in this paper, the multilayer perceptron was trained on only one specification, a mass flow of 20 g/s with a ratio of brick of 25 %, where we used the tracks of both brick and sand-lime brick for training. Images were captured at a frame rate of 100 Hz. The belt velocity was approximately 1 m/s.

The ground truth for the particle's arrival time and location was generated using the concept of a *virtual separation bar* (see [6,8]), since their exact values are not accessible due to the lack of a camera capturing the scene at the separation bar and the limited temporal resolution of most cameras. For this reason, only the images of the area-scan camera are used for training. Therefore, the prediction is performed with respect to a specific pixel row in the camera image corresponding to the virtual separation bar and the tracking phase is shortened accordingly. In addition, the coordinate system for the measurements is shifted so that the virtual separation bar coincides with the real one. The ground truth is then obtained by linear interpolation between the last measurement before and the first measurement after the virtual separation bar. For deployment, the trained network is applied to the original configuration and fed with non-shifted measurements. Although this concept introduces some inaccuracies due to interpolation errors and the assumption of similar particle motion on the belt and in the flight phase, it offers the benefit of not requiring additional sensors and allowing the network to be trained in an unsupervised fashion without additional costs for manually labeling the ground truth.

3.2 Experimental Results

The *true negative rate* (TNR) and *true positive rate* (TPR) were determined as performance indicators for the sorting quality. The TNR refers to the

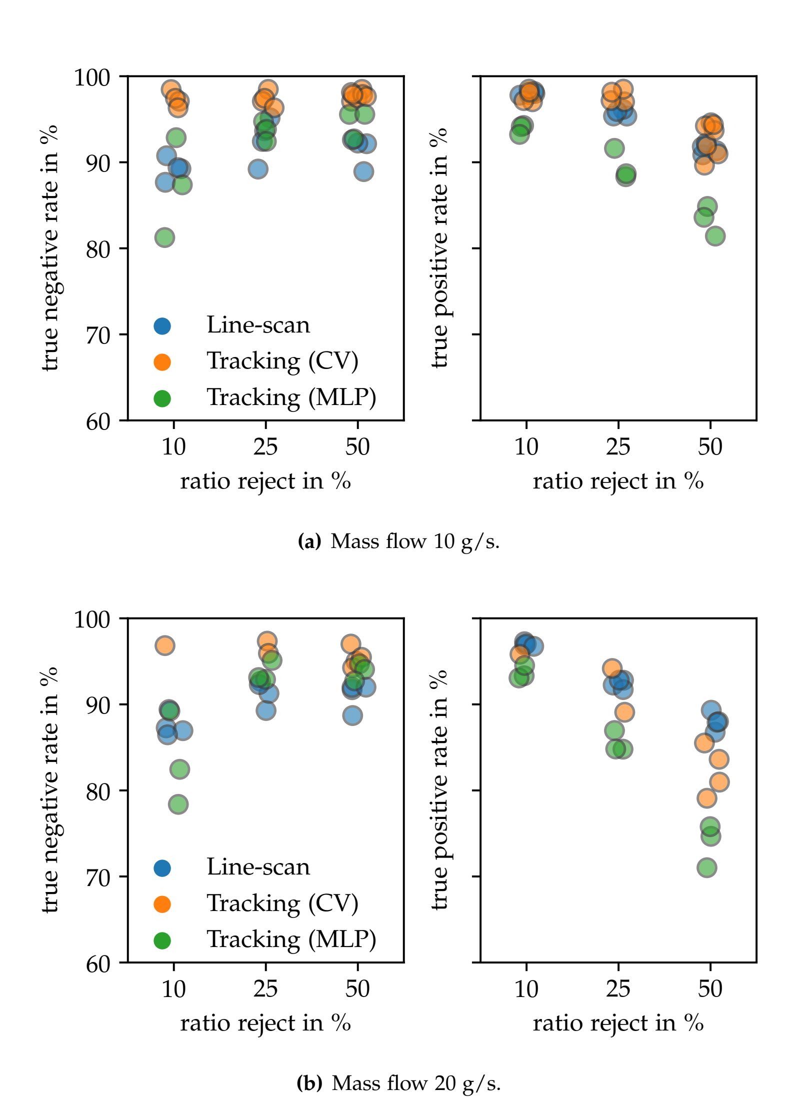
proportion of *residue* material that has been successfully removed, and the TPR to the proportion of *product* material that has successfully been accepted, i. e., not been removed. A selection of the results obtained is shown in Figure 4. The individual markers represent the result of an individual experiment.

As can be seen from Figure 4, the preliminary results show that the novel system achieves results comparable to a highly optimized Kalman filter-based one, although it does not outperform it. However, considering the early stage of development and the opportunities for increasing performance, e. g., by means of training data, we consider it a promising future research direction. An already gained benefit lies in avoiding tiresome manual tuning of parameters of the motion model, as the novel approach allows learning its parameters by provided examples due to its data-driven nature.

4 Conclusion

In this paper, we presented the experimental validation of a novel neural network-based multiobject tracking system. For this paper, we implemented and integrated the system for use with a laboratory-scale sorting system that was equipped with an area-scan camera. We compared the performance to ones achieved using a line-scan-based system as well as a multiobject tracking system with physically-motivated motion models. Preliminary results show that the novel system achieves results comparable to a highly optimized Kalman filter-based one, although it does not outperform it yet. However, an advantage of the novel system lies in avoiding tiresome manual tuning of parameters of the motion model.

Considering the early stage of development of the system, we believe there exist various interesting research directions to boost its performance. Great potential is believed to lie in the expansion and systematic selection of training data. Furthermore, a system combining physically-motivated as well as machine learning-based models as described in [6] is of great interest.



(a) Mass flow 10 g/s.

(b) Mass flow 20 g/s.

Figure 4: Results of the sorting experiments using the three different prediction models in terms of TNR and TPR. The individual markers represent the result of an individual experiment.

Acknowledgement

IGF project 20354 N of research association Forschungs-Gesellschaft Verfahrens-Technik e.V. (GVT) was supported by the AiF under a program for promoting the Industrial Community Research and Development (IGF) by the Federal Ministry for Economic Affairs and Climate Action on the basis of a resolution of the German Bundestag.

References

1. S. P. Gundupalli, S. Hait, and A. Thakur, "A review on automated sorting of source-separated municipal solid waste for recycling," *Waste Management*, vol. 60, pp. 56–74, Feb. 2017.
2. N. Dias, I. Garrinhas, A. Maximo, N. Belo, P. Roque, and M. T. Carvalho, "Recovery of glass from the inert fraction refused by MBT plants in a pilot plant," *Waste Management*, vol. 46, pp. 201–211, Dec. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0956053X15300684
3. B. Küppers, S. Schloegl, G. Oreski, R. Pomberger, and D. Vollprecht, "Influence of surface roughness and surface moisture of plastics on sensor-based sorting in the near infrared range," *Waste Management & Research*, vol. 37, no. 8, pp. 843–850, Jun. 2019. [Online]. Available: https://doi.org/10.1177/0734242X19855433
4. G. Maier, A. Shevchyk, M. Flitter, R. Gruna, T. Längle, U. D. Hanebeck, and J. Beyerer, "Motion-based visual inspection of optically indiscernible defects on the example of hazelnuts," *Computers and Electronics in Agriculture*, vol. 185, 2021.
5. G. Maier, F. Pfaff, C. Pieper, R. Gruna, B. Noack, H. Kruggel-Emden, T. Längle, U. D. Hanebeck, S. Wirtz, V. Scherer, and J. Beyerer, "Experimental evaluation of a novel sensor-based sorting approach featuring predictive real-time multiobject tracking," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1548–1559, 2021.
6. J. Thumm, M. Reith-Braun, F. Pfaff, U. D. Hanebeck, M. Flitter, G. Maier, R. Gruna, T. Längle, A. Bauer, and H. Kruggel-Emden, "Mixture of experts of neural networks and kalman filters for optical belt sorting," *IEEE Transactions on Industrial Informatics*, 2021.
7. S. Dittrich, V. Thome, J. Nühlen, R. Gruna, and J. Dörmann, "Baucycle–verwertungsstrategie für feinkörnigen bauschutt," *Bauphysik*, vol. 40, no. 5, pp. 379–388, 2018.

8. F. Pfaff, M. Baum, B. Noack, U. D. Hanebeck, R. Gruna, T. Längle, and J. Beyerer, "Tracksort: Predictive tracking for sorting uncooperative bulk materials," in *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2015, pp. 7–12.

Fast and comprehensive FPGA based BLOB analysis with the Hybrid-BLOB concept

Simon Wezstein[1,2], Michael Stelzl[1], and Michael Heizmann[2]

[1] MSTVision GmbH,
Im Weiherfeld 10, 65462 Ginsheim-Gustavsburg, Germany
[2] Karlsruhe Institute of Technology,
Institute of Industrial Information Technology,
Hertzstraße 16, 76187 Karlsruhe, Germany

Abstract In this contribution we show our approach for a feature rich and high speed BLOB analysis on FPGAs. For the Hybrid-BLOB concept we use a combination of a single-pass BLOB analysis and a double-pass labeling algorithm. We use Basler's VisualApplets for the implementation of the concept on their microEnable 5 frame grabbers. We achieve the extraction of the gray value data of the BLOBs at factor 14 higher frame rates compared to the naive labeling of the complete image. This is achieved by limiting the maximum BLOB size to 128×128 px, which speeds up the double-pass labeling algorithm. Our concept is targeted at low latency and high throughput demanding applications where BLOBs are small, like sensor based sorting or surface inspection.

Keywords Image signal processing, FPGA, BLOB analysis

1 Introduction

In image processing the term Binary Large Object (BLOB) analysis refers to the extraction of connected components of a binary image with posterior calculation of the component's features like area, circumference, etc. The features are often used to classify these components. They are often called objects, as in many applications single objects are

segmented and analyzed. In inspection tasks these algorithms may be used for the classification of single objects, e.g. into "accept" or "reject" classes or to divide the defects even further, for example into "dent", "scratch", etc.

In image processing Field Programmable Gate Arrays (FPGAs) are used if high throughput, low latency or energy efficiency is demanded. For example FPGAs are used directly in cameras for post processing of the sensor data. They are also used in special applications like sensor based sorting or surface inspection.

MSTVision developed an FPGA based sensor based sorting platform, which aims at minimum latencies [1]. Its logic is completely implemented with VisualApplets (VA). VA is a proprietary development platform by Basler (formerly Silicon Software) for FPGA image processing logic development, tailored for their frame grabbers and devices with embedded VA support [2]. The platform proved its low latencies of around 200 μs in [3]. Currently the system's image processing capabilities are limited by the feature limits of the VA BLOB analysis operator.

To run the mentioned tasks on FPGAs, implementations of the BLOB analysis are required. The research field in FPGA based BLOB analysis algorithms is still active. To extract BLOB features, first the connected components need to be extracted. This process is named labeling, its output is an intermediate image, with unique pixel values for each connected component in the image. There are many algorithms, but the algorithms may be divided in four categories [4, p. 352-359]:

1. Single-pass algorithms, where the data only needs to pass the computing pipeline once.

2. Double-pass algorithms, where the data needs to pass the computing pipeline twice.

3. Multi-pass algorithms, where the data needs to pass the computing pipeline multiple times, depending on the image content.

4. Random-access algorithms, where the data needs to be accessed randomly.

Each algorithm category poses its own pros and cons. Most of the current research focuses on single-pass algorithms, as they provide the

lowest possible latencies and demand only small FPGA resource quantities. The BLOB labeling is done only implicit. The downside is the limited amount of extractable features, which we will explain in the next paragraph. We will focus on single- and double-pass algorithms, as they are used in this contribution.

1.1 Labeling problem in detail

The main problem for image stream labeling algorithms are "U" shaped components, for example see fig. 1. While processing the binary image stream, the first object pixel is observed at (1,4). A new label is created for a unique representation of the object. In the next image line at (2,1) another object pixel is observed, but based on the processed data, it's not connected with the ones in the line before. A new label is created. While scanning the line, both labels coexist. In line 3 both labels turn out to be connected at (3,3) or (3,4), depending whether the 4-connected or 8-connected neighborhood is used. This results in a problem: the previously assigned labels need to be merged into one. The way the algorithms overcome that problem is their fundamental difference.

Double-pass algorithms like [5, p. 4] use equivalence tables to record these conflicts. One object may consist of many intermediate labels. After the first labeling pass, a conflict resolving algorithm is used to convert the labels to a unique final label lookup table (LUT). With the LUT and the result image of the first pass, the final label image is created. The advantage over single-pass algorithms is the ability to extract the component pixel accurately. This enables the calculation of all object features after labeling. The disadvantages are their higher memory demands for buffering the intermediate label image and the equivalence table. Resolving the label conflicts and calculation of the features after the labeling adds computing time. For FPGA implementations the often required random memory accessibility for the equivalence table is a limitation, too.

Single-pass algorithms like [6] don't provide a label image output, instead they calculate the object features directly. The labeling is only carried out internally. A single-pass algorithm performing the extraction of the object area would work on the example in fig. 1 as follows: the first object pixel is observed at (1,4), a new temporary label and ac-

cumulator is created. For each connected pixel the area is incremented by 1. In the next image line at (2,1) another object pixel is observed, another temporary label and accumulator is created. When both labels collide, one label is deleted and its area accumulator is added to the other accumulator. The output of the algorithm is a list of component features, in this example only the area. There is no ability to extract the object pixels to compute other features. The features which may be extracted are limited to those which may be merged out of the values of sub component features on label collision. Their advantages are the small memory requirements, which is limited to the feature and label table, and the smaller computing time.

With single-pass algorithms features like the oriented bounding box or minimum/maximum Feret diameters can't be calculated. These features are usually calculated with the object's convex hull and the rotating calipers algorithm. [7]

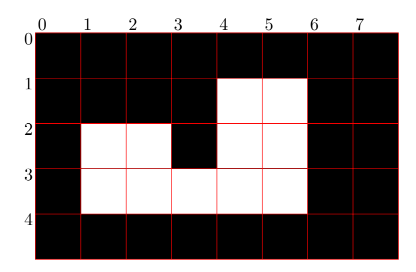


Figure 1: A simple "U" shaped object to demonstrate the main challenge of streaming labeling algorithms. Modified version from [8, Fig. 3]

2 Method

To fill the gap between single- and double-pass algorithms, we developed the Hybrid-BLOB concept. The method consists of two BLOB analysis/labeling algorithms, a single-pass algorithm and a double-pass algorithm. The single-pass algorithm is the one used in the VA BLOB analysis operators [9]. The double-pass algorithm is our implementation of the algorithm described in [8].

The double-pass algorithm is expensive with respect to computing

time and memory if applied to a big image. For big images, the conflict resolve table won't fit into the FPGA's on-chip memory of current Basler frame grabbers, requiring utilizing the off chip DRAM. Resolving the conflicts is an algorithm of quadratic order. The single-pass algorithm in comparison does only provide a few features.

To overcome the limitations, both algorithms are combined, as described in the next subsection. This allows smaller input image sizes for the double-pass algorithm, thus the conflict resolve table fits into the FPGA's on-chip memory and the conflict resolve algorithm may run faster.

2.1 Architecture overview

In fig. 2 the concept is shown. The image input is preprocessed and segmented. The single-pass BLOB analysis of VA is applied to the segmented image. In parallel, the segmented image and the preprocessed gray image are stored into dynamic random access memory (DRAM). A pre-classification is applied to the output of the single-pass BLOB analysis. The remaining objects of interest are retrieved from the DRAM buffer via their bounding box information. The extracted object images may contain pixels of other objects, as shown in the example BLOBs in fig. 2. The double-pass algorithm is then used to label the small images. With the bounding box information of the previous BLOB analysis and the label image, the corresponding object may be extracted from the binary and gray image. Afterwards we extract various object features which then may be used for object classification.

2.2 Implementation

The implementation is done in VA with only VA operators except one VHDL custom operator. The target hardware platform are the microEnable 5 marathon frame grabbers, [11]. As the implementation of most of the single architecture elements is straightforward, we focus on the double-pass labeling algorithm and the feature extraction. For comparison we use an implementation of the labeling algorithm for the labeling of a whole 1024×1024 px image. The maximum configurable bounding box size for labeling in Hybrid-BLOB is limited to

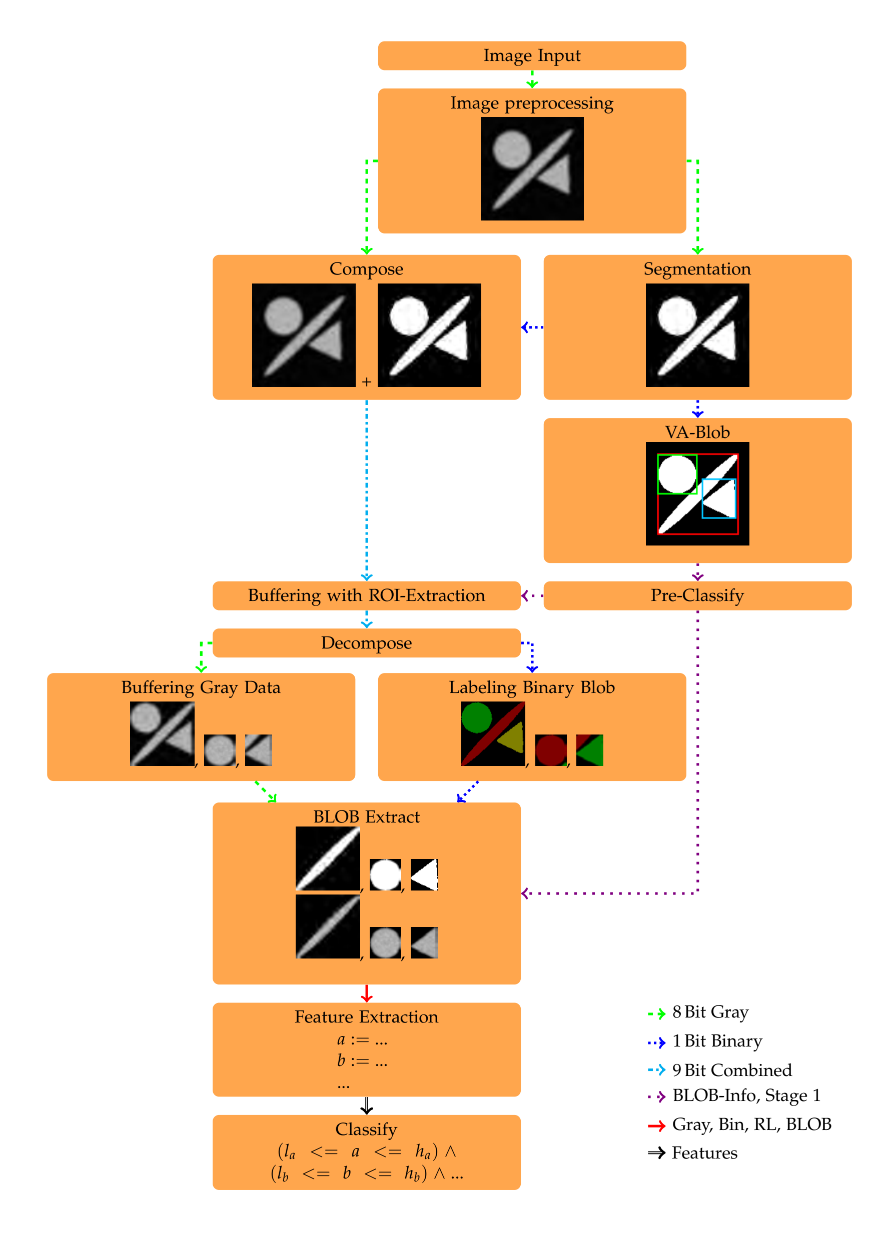


Figure 2: Hybrid-BLOB architecture overview. Modified version from [10, Fig. 4.1]

Table 1: Comparison of memory requirements of the implementations. The maximum code length was empirical determined. *id* is the label id, *eq* is the equivalent label id in case of a conflict, *r* is the run element's row, *s* and *e* are start and end column of the run. *BRAM* is for Block Random Access Memory, the FPGA's on-chip memory. [10, Tab. 4.1]

Parameter	Labeling	Reduced Labeling
id, eq	16 Bit	8 Bit
r, s, e	13 Bit	7 Bit
Memory per element	10 Byte	5 Byte
Max. label count	65535	255
Max. code length	65535	4095
Maximum memory	5.24 MBit	163.8 kBit
BRAM-Elements (18 kiB per element)	291.3	9.1

128×128 px. The labeling stage uses fixed frame size inputs of the configured maximum bounding box size. The design transfers the input image and an image with the BLOB features over Direct Memory Acess (DMA) channels.

Labeling The labeling algorithm is an implementation of [8]. The algorithm is a run length encoding (RLE) based, 4 connected neighbourhood type. Depending on the design, other bit depths are used for the labels and the run length code. Labeling smaller images results in smaller coordinate bits and fewer possible labels. In tab. 1 the resource occupation for both variants are shown. By reducing the image size which has to be labeled, the required memory drops, which practically enables the storage of the run length data in the FPGA's on-chip memory. For the labeling of the whole image, the data is stored in the frame grabber's DRAM. The whole image labeling design does not contain the calculation of features.

Feature extraction With the extracted object's image data, the feature extraction takes place. The extraction is completely integrated into the FPGA. The orientated bounding box and Feret features are not calculated with the convex hull and rotating calipers. They are approximated by discrete object rotations in angle steps of 0.703 °. To save FPGA resources, the calculation of unneeded features may be removed.

The extracted features are:

- VA-Operator: *bounding box, area, center of gravity* (output of single-pass analysis stage).
- Gray Value: *min, max, mean, std, median, upper and lower quartile, difference to a reference histogram (rel/abs)*.
- Other binary image features: *Euler's number, circumference, compactness, circularity, circle equivalent diameter*.
- Binary image moments: *2nd and 3rd order*.
- Ellipse features: *main axis angle, main and minor axis radius, eccentricity*.
- Gray image moments: *2nd and 3rd order*.
- Oriented bounding box: *area, angle, width, height*.
- Feret diameter: *minumum, maximum, min. angle, max. angle*.

For further information about the features, we suggest [12], [13], [14], [15] and [16].

3 Results

Both test designs have been built with VisualApplets 3.3.1 for the microEnable 5 Marathon VCLx frame grabber, running at a frequency of 125 MHz [17] [18]. The used frame grabber runtime is version 5.7. In fig. 3 our test image is shown. It contains 1161 BLOBs and has a resolution of 1024×1024 px. The amount of objects is not representative for real applications. The image is uploaded to the FPGA DRAM and repeatedly processed for our tests. We use the shown frame rate of microDisplay, the runtime application used to configure the frame grabber. The frame rates are validated against debug registers on the FPGA. The BLOB frequency is measured with a debug register. For our measurements, we don't filter the output of the single-pass stage.

The BLOB count varied from 1143 to 1161 while testing. The reason of these variations is currently unknown, their origin is the single-pass

Table 2: Measurement results for processing the image shown in fig. 3.

Parameter	Full Labeling	Hybrid-BLOB
Frame rate	1.25 Hz	17.0 ± 0.5 Hz
Mean BLOB frequency	1.5 kHz	19.8 ± 0.010 kHz
Mean time per BLOB	689 μs	51 μs
Labeling throughput	1.3 Mpx/s	324.4 Mpx/s

stage. We use 1161 as BLOB count for calculating the mean time of the labeling design and the period of the BLOB frequency for the Hybrid-BLOB design. In tab. 2 our results are shown. Our Hybrid-BLOB concept runs at 14 times higher frame rates even with the 250 times higher data throughput in the double-pass stage. Due to the fixed frame size for bounding box extraction, the labeling overhead increases if many small objects are present.

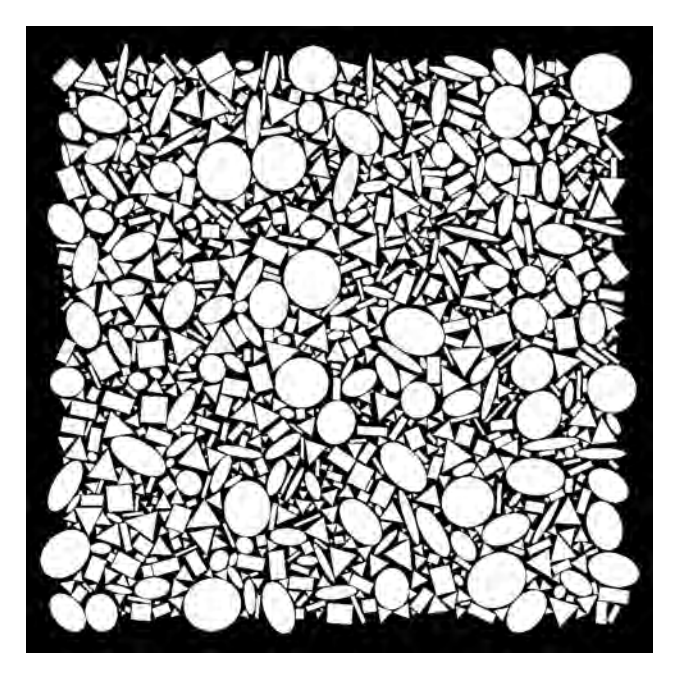


Figure 3: Test image used. It contains 1161 objects at a resolution of 1024×1024 px.

4 Conclusion

We have shown an approach to speed up a feature rich BLOB analysis on FPGAs. The implementation with VisualApplets enables the usage on Basler frame grabbers of the current portfolio and possible future platforms supporting VisualApplets. Hybrid-BLOB processes in our test scenario 19.800 BLOBs per second, which allows its usage in the field of granule sorting. To increase the throughput further, the labeling and feature extraction stage may be implemented multiple times in parallel. Our concept may be used in traditional PC based image processing, too.

The throughput and latency may be further improved if the double-pass labeling algorithm is extended to support variable image input sizes for overhead reduction. If variable input sizes are used, the run length encoding stage runs faster and the count of runs to label decreases. We expect big improvements if small BLOBs are processed, as the measured overhead is 250 times compared to the image input.

Acknowledgments

The work described here results from the project "Hybride Bildverarbeitung auf FPGAs", supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag, for which we are grateful. We want also to thank Basler AG for giving the opportunity for the work described in [10], which is the origin of our concept.

References

1. MSTVision GmbH, "Minimale Responsezeit," https://mstvision.de/downloads-sorting/, 2020, online, accessed 25-September-2022.
2. Basler AG, "VisualApplets Graphic FPGA Programming," https://baslerweb.com/en/products/visualapplets/, 2022, online, accessed 15-September-2022.
3. S. Wezstein, M. Stelzl, and M. Heizmann, "Latency evaluation of an FPGA-based sorting system," in *9th Sensor-Based Sorting & Control 2022*, Greiff, Kathrin and Wotruba, Hermann and Feil, Alexander and Kroell, Nils and

Chen, Xiaozheng and Gürsel, Devrim and Merz, Vincent, Ed., 04 2022, pp. 143–160.

4. D. G. Bailey, *Design for Embedded Image Processing on FPGAs*. Singapore: John Wiley & Sons (Asia) Pte Ltd, 2011.
5. Y. T. Kong and A. Rosenfeld, *Topological Algorithms for Digital Image Processing*. Amsterdam, The Netherlands: Elsevier Science B.V., 1996.
6. D. Bailey and C. Johnston, “Single pass connected components analysis,” *Proceedings of Image and Vision Computing*, 01 2007.
7. G. Toussaint, “Solving geometric problems with the rotating calipers,” in *Proceedings of MELECON ’83, Mediterranean Electrotechnical Conference*, 1983.
8. K. Appiah, A. Hunter, P. Dickinson, and J. Owens, “A run-length based connected component algorithm for FPGA implementation,” in *2008 International Conference on Field-Programmable Technology*, Dec 2008, pp. 177–184.
9. Basler AG, “Library Blob,” https://docs.baslerweb.com/visualapplets/files/manuals/content/library.Blob.html, 2022, online, accessed 16-September-2022.
10. S. Wezstein, “FPGA basierte Blobanalyse mit dem „Hybrid-BLOB”-Verfahren,” Master’s Thesis, Hochschule Darmstadt, 2021.
11. Basler AG, “microEnable 5 marathon Frame Grabber Portfolio,” https://www.baslerweb.com/en/products/acquisition-cards/microenable-5-marathon/, 2022, online, accessed 15-September-2022.
12. B. Jähne, *Digitale Bildverarbeitung*, 7th ed. Berlin: Springer Vieweg, 2012.
13. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. USA: Thomson-Engineering, 2007.
14. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, fourth edition, global edition ed., Array, Ed. New York, NY: Pearson Education Limited, 2018.
15. J. Ohser, *Angewandte Bildverarbeitung und Bildanalyse*. Leipzig: Carl Hanser Verlag GmbH & Company KG, 2018.
16. R. M. Haralick and L. G. Shapiro, *Computer And Robot Vision*. USA: Addison-Wesley Publishing Company, Inc., 1992.
17. Basler AG, “VisualApplets 3 User Manual,” https://docs.baslerweb.com/visualapplets/files/manuals/content/device%20resources.html, 2022, online, accessed 15-September-2022.
18. ——, “microEnable 5 marathon VCLx - Frame grabber,” https://www.baslerweb.com/en/products/acquisition-cards/microenable-5-marathon/microenable-5-marathon-vclx/, 2022, online, accessed 15-September-2022.

An introduction to quantum image processing on real superconducting quantum computers

Alexander Geng[1,2], Ali Moghiseh[2], Katja Schladitz[2], and Claudia Redenbach[1]

[1] University of Kaiserslautern, Gottlieb-Daimler-Straße 47, 67663 Kaiserslautern
[2] Fraunhofer Institute for Industrial Mathematics ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern

Abstract The size of images and data we process every day have been growing exponentially over the last years. Quantum computers promise to process this data more efficiently. Experiments on quantum computer simulators prove the paradigms this promise is built on to be correct. However, currently, running the very same algorithms on a real quantum computer is often too error prone to be of any practical use. We explore the current possibilities for image processing on real quantum computers. We redesign a commonly used quantum image encoding technique to reduce its susceptibility to errors. We show experimentally that the current size limit for images to be encoded on the quantum computer and subsequently retrieved with an error of at most 5% is 2×2 pixels. A way to circumvent this limitation is to combine ideas of classical filtering with a quantum algorithm operating locally, only. We show the practicability of this strategy using the application example of edge detection. Our hybrid filtering scheme's quantum part is an artificial neuron, working well on real quantum computers, too.

Keywords Quantum image processing, quantum image encoding, quantum edge detection, quantum artificial neurons, IBM Quantum Experience

1 Introduction

In this contribution, we do not discuss quantum imaging methods. Throughout, we assume the image data to be processed on a quantum computer to be given as a classical gray-value image. Thus, first, we have to encode the gray-value information into quantum states. There are basically three concepts for this encoding, namely basis encoding, phase encoding, and amplitude encoding. Within the last years, several methods have been developed following these three basic concepts [1]. Here, we concentrate on the phase encoding method Flexible Representation of Quantum Images (FRQI) [2] and improve its implementation.

After the encoding, we normally process the states by applying some algorithms. Initially, algorithms were only formulated in theory or executed on simulators of quantum computers. Only since 2016, it has also been possible to execute algorithms on real quantum computers. A short overview of currently available algorithms is given in [3]. Here, we aim at algorithms that run on the actual quantum hardware. More precisely, we implement quantum image processing algorithms on IBM's superconducting quantum computers [4].

This paper is organized as follows. Section 2 provides some basics of quantum computing. In Section 3, we describe the experimental setup including the quantum computers, the software, and the classical computers used. We explain our improved version of the FRQI image encoding in Section 4. In Section 5, we present the idea of hybrid quantum image filtering and highlight the performance for detecting edges in images with a quantum computer. Two variants of the quantum edge detector with 2D and 1D masks are detailed. Section 6 concludes the paper.

2 Quantum computing basics

Before diving into quantum image processing, we summarize some basic concepts of quantum computing [5]. Classical computing and quantum computing follow completely different paradigms, starting with the basic elements. Classically, everything builds on bits, that can attain either state 0 or 1. The quantum analogue are the quantum bits (qubits) – two-state quantum systems that allow for more flexibility.

Analogous to 0 and 1, there are two basis states of a qubit: $|0\rangle = (1,0)^T$ or $|1\rangle = (0,1)^T$. However, any linear combination (superposition)

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

of the basis states with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$ defines a possible state, too. The overall phase of a quantum state is unobservable [5]. That is, $|\psi\rangle$ and $e^{i\xi}\,|\psi\rangle$ for $\xi \in [0, 2\pi]$ define the same state. Hence, it is sufficient to consider $\alpha \in \mathbb{R}$.

As a consequence, the state of a single qubit can be visualized as a point on the unit sphere in $\mathbb{R}^3$ (Bloch sphere) with spherical coordinates ϕ and θ, where $\alpha = \cos(\theta/2)$ and $\beta = e^{i\phi}\sin(\theta/2)$. All operations on a qubit must preserve the condition $|\alpha|^2 + |\beta|^2 = 1$, and can thus be represented by 2×2 unitary matrices. Standard operations (so-called gates) acting on single qubits are

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}, \tag{2}$$

where the X-gate acts like a classical NOT operator and the Hadamard gate (H) superposes the basic states of a single qubit. A qubit in superposition can be thought of as having all possible states at the same time. The Phase gate (P) rotates by θ about the z-axis of the Bloch sphere. Phase shift gates can be used to encode gray-values.

Additionally, we need operations that link two or more qubits. The most common operation in quantum computing is the controlled NOT-gate (CX-gate) taking two input qubits. The target qubit's state is changed depending on the state of the control qubit:

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3}$$

That means, if the control qubit is in state $|1\rangle$, then we apply an X-gate to the target qubit. Otherwise, we do nothing. For example, assume our two qubit system has the state $|10\rangle = |1\rangle \otimes |0\rangle$, where the first qubit is the control, the second the target qubit and $\otimes$ is the tensor product. Then, application of the CX-gate results in the state

$$|11\rangle = |1\rangle \otimes |1\rangle = (0,0,0,1)^T. \tag{4}$$

So basically, the application of quantum gates can be formulated in terms of linear algebra.

In general, we can apply any unitary operation to the target qubit. For example, a controlled-Phase gate applies a P-gate to the target qubit if and only if the control qubit is in state $|1\rangle$. We can also increase the number of control qubits even further. The operation with two control qubits and an X-gate applied to the target qubit is called Toffoli gate.

Applying such controlled operations to two or more qubits with the control qubits in superposition, results in the entanglement of the qubits involved. In terms of linear algebra, an entangled state of several qubits is one that cannot be written as a tensor product of states of the individual qubits. Entanglement is exactly where we benefit from the quantum computing properties. Together with superposition, entanglement allows to use a logarithmically lower number of qubits compared to the number of classical bits.

While in a classical computer all bits are connected to each other, in IBM's quantum computer the qubits are arranged in a special, so-called heavy-hexagonal scheme (see the honeycomb structure in Figure 1). That is, each qubit is directly connected to at most three other qubits. To apply two qubit gates to unconnected qubits, the information has to be swapped to neighbouring qubits by application of additional CX-gates. Each CX-gate, however, increases the overall error considerably such that an algorithm should employ as few CX-gates as possible.

Lastly, the readout is also completely different for classical and quantum computing. On classical computers, you can always read the current state of the bits, copy them, or just continue running an algorithm with the same state of the bits as before the readout. Unfortunately, this is not possible on quantum computers. First, according to the no-cloning theorem [5], a state cannot be copied. Second, when measuring (reading out the state of) a qubit, its state collapses to one of the basis states $|0\rangle$ or $|1\rangle$. Hence, continuing the algorithm after read out is not possible. Additionally, measuring a qubit does not immediately provide the values of α and β in Equation (1). However, the probability of collapsing to $|0\rangle$ is given by $|\alpha|^2$ while the state $|1\rangle$ is obtained with probability $|\beta|^2$. Repeated measurements (shots) of the same state allow for an estimation of these probabilities and thus the values α and β, too. For further reading on quantum computing basics we recommend [5].

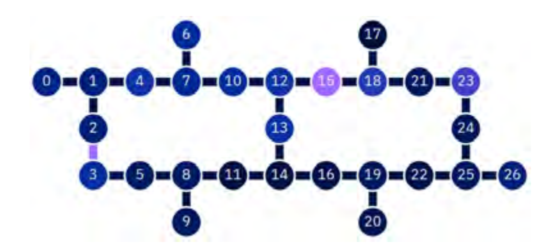


Figure 1: Coupling map of the backends used in this paper. Every circle represents a qubit, lines represent connections between the qubits. Colors code the readout errors (circles) and the CX-errors for the connections (lines). Dark blue indicates a small error, purple a large one. Errors are shown for 'ibmq_ehningen'. 'ibmq_toronto' has the same coupling map, but errors differ slightly (see Table 2).

3 Near-term quantum computers

We use the open-source software development kit Qiskit [6] for working with IBM's circuit-based superconducting quantum computers [4]. They provide a variety of systems, also known as backends, which differ in the type of the processor, the number of qubits (scale), and their connectivity [4]. Access is provided via a cloud. In this paper, we use two of the available 22 backends, 'ibmq_toronto' and 'ibmq_ehningen' see Table 1. This choice is not crucial for our use case as we use a small subset of the qubits only and backends' performance does not differ significantly. The coupling map, so the connections between the qubits, of the backend 'ibmq_ehningen' is shown in Figure 1. Additional parameters describing the performance of IBM's backends are quality (quantum volume) and speed (circuit layer operations per second, CLOPS). All parameters of the two used backends are summarized in Table 1.

Besides the coupling map and the above listed performance values, external conditions influence the backends. Thus, compared to classical computers, the basic operations of quantum computers yield quite large errors. E. g., applying a couple of gates or performing measurements is currently quite noisy with errors that can change hourly. Typical average values for CX error, single qubit gate error, and readout error, are shown in Table 2. Additionally, Table 2 shows the decoher-

Table 1: Processor type and actual performance of the used backends as measured in September 2022.

Backend	Processor type	Scale [# qubits]	Quality [QV]	Speed [CLOPS]
'ibmq_toronto'	Falcon r4	27	32	2.800
'ibmq_ehningen'	Falcon r5	27	64	1.900

Table 2: Typical average calibration data of the two chosen backends. The values are from September 2022.

Backend	CX-error [%]	Single qubit gate error [%]	Readout error [%]	T1 [μs]	T2 [μs]
'ibmq_toronto'	5.34	0.051	3.66	103.71	107.72
'ibmq_ehningen'	0.71	0.024	1.05	151.74	160.92

ence times T1 – a decay constant measuring, how probable a qubit stays in the state $|1\rangle$ and not $|0\rangle$, and T2 – the dephasing time measuring how long the phase of a qubit stays intact. The circuit depth counts the maximal number of basis operations performed by a single qubit during an algorithm. A high circuit depth will result in an accumulation of errors during the runtime of the algorithm.

An additional issue in quantum computing is that only a few operations, called basis gates, can be performed on the quantum computer. Currently, IBM's superconducting quantum computers have five basis gates: the identity, X-, CX-, and P-gates, and the square root X (SX-)gate rotating by $\pi/2$ about the x-axis of the Bloch-sphere [4]. Qiskit includes a transpiler, which decomposes a given algorithm into these basis gates and optimizes these steps in some way [6]. Nevertheless, keeping the available basis gates in mind when developing algorithms helps to limit their overall number.

For preparing data and generating and storing the circuits before sending them to the quantum computer, we use a classical computer with an Intel Xeon E5-2670 processor running at 2.60 GHz, a total RAM of 64 GB, and Red Hat Enterprise Linux 7.9.

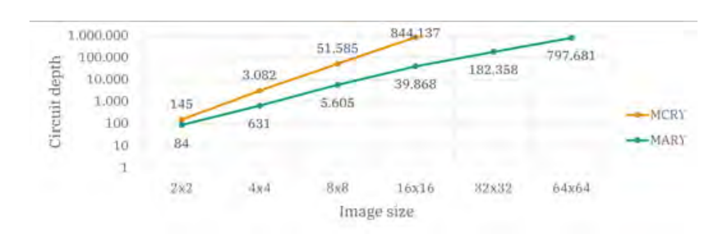


Figure 2: Circuit depth for varying image sizes and MCRY-/MARY-implementation on backend 'ibmq_toronto'. Mean values of 10 observations in logarithmic scale.

4 Quantum image encoding

There are many methods for encoding images in quantum computers. One of the most frequently mentioned methods is FRQI introduced in [2]. Assume that we want to encode a $2^n \times 2^n$ pixel gray-value image. We split the required qubits into two parts - $2n$ qubits for the pixel positions and one qubit for the gray-value information. Practically, FRQI can be implemented on superconducting quantum computers by using entanglement between the position qubits and the gray-value qubit. We take a closer look at the heart of the FRQI algorithm, the multi-controlled y-rotation gate (MCRY). It applies a rotation around the y-axis corresponding to the gray-value only if all position (aka control) qubits are in state $|1\rangle$. Subsequently, we change the state to which the actual phase should be applied by X-gates. Thus,in total we need one MCRY gate for each gray-value in the classical image. As discussed above, on a real backend, complex operations like MCRY have to be constructed by concatenating available basis gates.

Inspired by [7], we replace MCRY by what we call multi-adapted-controlled y-rotation gates (MARY). Our MARY gates need less basis gates, especially less of the particularly error-prone CX-gates. Thus, the replacement reduces the overall error significantly. Moreover, fewer gates and lower circuit depth (Figure 2) speed up calculations. The impact of replacing MCRY by MARY increases with image size. In MCRY, all qubits would ideally have to be connected with each other. Hence, missing connections on the real backends have to be circumvented by swapping with CX-gates. In contrast, MARY requires a much smaller connectivity between the qubits.

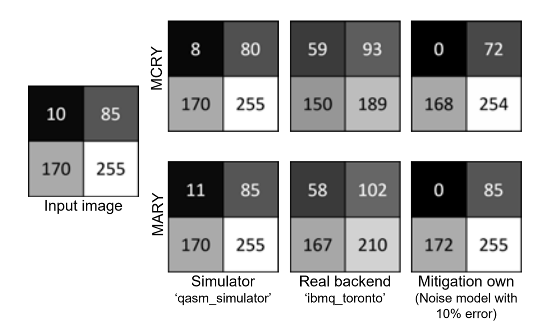


Figure 3: Results for 2×2 gray-value images using the mean of the executions. In the last column, some measurement error mitigation techniques have been applied [8].

Figure 3 shows the performance on a 2×2 sample image. The hardware induced error is clearly visible in the results achieved on the real backend. In fact, there, we can only retrieve the image with acceptable error when applying measurement error mitigation [8]. That is, from observations on exactly this backend, the distribution of the error is estimated. Inversion of the error model then improves the results. To our knowledge, image retrieval with FRQI for images larger than 2×2 is currently not possible on real backends, see also [8–10].

Table 3 shows our findings for the maximum *executable* and *usable* image sizes for the MCRY- and MARY-implementations. Executable here means, it is possible to run the algorithm at all without focusing on the outcomes. Usable implies that the relative difference between input image and reconstructed image is less than 5%. We clearly see a benefit of the MARY-implementation in terms of maximum *executable* image size. However, due to the high noise level of the backends, we could not increase the maximum *usable* image size.

Having experienced this tight restriction, we still aim at image processing algorithms which are robust to the hardware noise in the current noisy intermediate-scale (NISQ) era and hence executable on the real backends. In the next section, we describe a design pattern for algorithms meeting these demands.

Table 3: Current maximum executable and usable image sizes for MCRY- and MARY-implementations on 'qasm_simulator' with 8.192 shots and IBM's backend 'ibmq_toronto' limited to 64 GB memory.

	maximum executable image size		maximum usable image size	
Method	'qasm_simulator'	'ibmq_toronto'	'qasm_simulator'	'ibmq_toronto'
MCRY	32×32	16×16	16×16	2×2
MARY	256×256	32×32	16×16	2×2

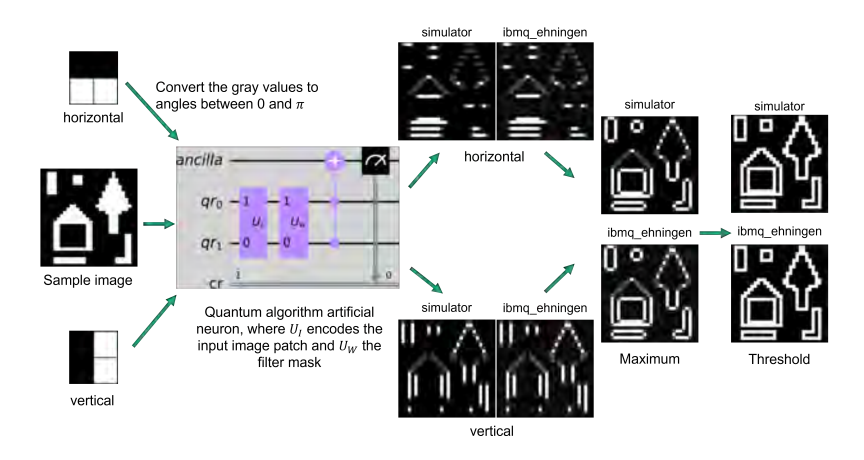


Figure 4: Scheme from [12] for edge detection in a 30×30 sample image by using 2×2 filter masks, 'qasm_simulator' and backend 'ibmq_ehningen' (executed on October, 15 2021) with 8.192 shots, and ToolIP [13] for post-processing.

5 Quantum image filtering

In this section, we introduce a class of hybrid algorithms combining classical filtering with quantum computing on 2 x 2 pixel patches. As an example, we combine classical edge detection with a quantum artificial neuron [11] as sketched in Figure 4. We calculate the inner product of the input image patch and the filter mask not only on a simulator but also on real quantum computers [12]. Being restricted to 2 x 2 masks, we can either apply that directly or split our task into one-dimensional filtering steps. The latter is more robust with respect to noise [12].

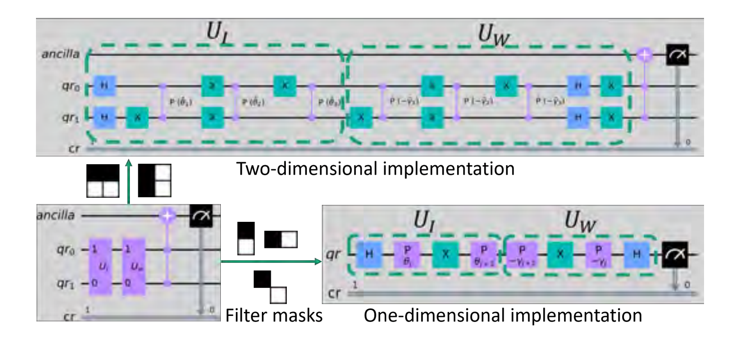


Figure 5: Hybrid quantum edge detection. U_I encodes the input image patch and U_W the filter mask. The gray value information is encoded in the P-gates. In the 1D case, the additional diagonal direction is required for detecting corners, too.

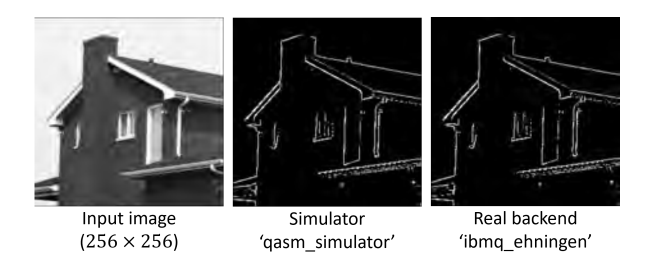


Figure 6: Results for the 256 × 256 House image [14]. The 'qasm_simulator' and backend 'ibmq_ehningen' results differ only slightly.

Moreover, only a very small number of gates and only one qubit per direction and pixel are required. This ensures that a very small number of shots (measurements) suffices for identifying the edges of the image. The lower number of shots in turn reduces the execution time significantly. The quantum circuits of the two implementations are shown in Figure 5.

Figure 6 shows the results of our hybrid 2D edge detection for a typical toy example image [14]. In [12], we process 256 × 256 pixels gray-value images. Further extension to larger images increases the number of circuits, only, but does not decrease the robustness of our algorithm. Nevertheless, in the end, we create one circuit for each combination of

input image patch and filter mask. This can scale up quite fast with larger images. In classical computing, this can be compensated by parallelization. In fact, this is also an option in quantum computing. We can use several qubits in parallel and process multiple image patches at the same time. By that, we decrease the number of needed circuits and also the execution time in the end. Mid-circuit measurement [4] allows to measure a qubit at any step of the algorithm and use the same qubit again for further calculations.

6 Conclusion

Quantum computing is potentially very useful in image processing. It promises exponentially lower memory usage in terms of qubits compared to classical bits and also faster calculations. However, the currently available noisy intermediate-scale quantum computers are still quite error-prone and hardware improvement is subject of vividly ongoing research. At the moment, image retrieval is only possible for images up to a size of 2×2. A strategy to deal with these limitations is to combine quantum and classical algorithms. In such hybrid solutions, the quantum computing part is actually much smaller than the classical part. We use only a small number of gates, and avoid or decrease the number of particularly error-prone types. The quantum computing share can be extended gradually along with the hardware progress. Instead of trying to implement all image processing functionality on quantum computers, we should rather identify, for which problems and which steps in complex algorithms quantum computing can be helpful or eventually even beat classical machines.

Acknowledgement

This work was supported by the project AnQuC-3 of the Competence Center Quantum Computing Rhineland-Palatinate (Germany).

References

1. F. Yan, A. M. Iliyasu, and S. E. Venegas-Andraca, "A survey of quantum image representations," *Quantum Information Processing*, vol. 15, no. 1, pp.

1–35, 2016, https://doi.org/10.1007/s11128-015-1195-6.

2. P. Q. Le, F. Dong, and K. Hirota, "A flexible representation of quantum images for polynomial preparation, image compression, and processing operations," *Quantum Information Processing*, vol. 10, no. 1, pp. 63–84, 2011, https://doi.org/10.1007/s11128-010-0177-y.
3. K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, "Noisy intermediate-scale quantum algorithms," *Reviews of Modern Physics*, vol. 94, no. 1, p. 015004, 2022.
4. IBM, "IBM Quantum," 2022, https://quantum-computing.ibm.com/. Accessed September 2022.
5. M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*. New York: Cambridge University Press, 2000.
6. H. Abraham *et al.*, "Qiskit: An Open-source Framework for Quantum Computing," 2019, https://doi.org/10.5281/zenodo.2562110.
7. H. Co, E. Peña Tapia, N. Tanetani, J. P. Arias Zapata, and L. García Sanchez-Carnerero, "Quantum imaging processing (a case study: cities at night)," gitHub repository, https://github.com/shedka/citiesatnight. Accessed June 10, 2021.
8. A. Abbas *et al.*, "Learn quantum computation using Qiskit," 2020, https://qiskit.org/textbook/. Accessed September 2022.
9. M. Harding and A. Geetey, "Representation of Quantum Images," 2018, https://www.cs.umd.edu/class/fall2018/cmsc657/projects/group_6.pdf.
10. G. Cavalieri and D. Maio, "A quantum edge detection algorithm," 2020, preprint at https://arxiv.org/abs/2012.11036.
11. S. Mangini, F. Tacchino, D. Gerace, C. Macchiavello, and D. Bajoni, "Quantum computing model of an artificial neuron with continuously valued input data," *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045008, 2020, https://doi.org/10.1088/2632-2153/abaf98.
12. A. Geng, A. Moghiseh, C. Redenbach, and K. Schladitz, "A hybrid quantum image edge detector for the NISQ era," *Quantum Machine Intelligence*, vol. 4, no. 15, 2022, https://doi.org/10.1007/s42484-022-00071-3.
13. Fraunhofer Institute for Industrial Mathematics, "ToolIP - tool for image processing," www.itwm.fraunhofer.de/toolip. Accessed September 2022.
14. A. Sawchuk *et al.*, "House 4.1.05," USC-SIPI image database, 1973, http://sipi.usc.edu/database/. Accessed September 2022.

High-performance image reconstruction algorithm in CUDA C++ for ultra wideband multi-channel MIMO radar systems

Josh Perske, Harun Cetinkaya, Christopher Schwäbig, and Sabine Gütgemann

Fraunhofer Institute for High Frequency Physics and Radar Techniques FHR, Fraunhoferstraße 20, 53343 Wachtberg

Abstract The exact measurement of process-relevant parameters and product properties are prerequisites for efficient and sustainable production. In addition to accuracy, industrial applications place tough demands on the real-time capability and achievable measurement rates of the sensor technology. In the past, radar signal processing was mainly done with the use of highly specialised hardware to achieve the necessary performance. Computer systems are used to perform simulations and to test new algorithms before being implemented under high effort. The resulting sensor systems are rigid, and their enhancement is time and cost consuming. With increasingly powerful graphics processing units (GPU) and the possibility to use them for general-purpose computing, a new approach is to outsource parts of the radar signal processing from the specialised hardware to commercially available computer systems. The main objective of this idea is to reduce the development time of new sensor systems, facilitate their modification and to increase the re-usability of produced code. This approach is tested with a new imaging radar algorithm, developed for a frequency modulated continuous wave (FMCW) radar system with a modular multiple input multiple output (MIMO) antenna array. The implementation of this algorithm is used to determine the boundaries of this new approach and involves a step-by-step optimisation process to improve the performance of the final result.

Keywords Imaging radar algorithm, FMCW radar, MIMO sensor system, back-projection, CUDA C++, GPU programming

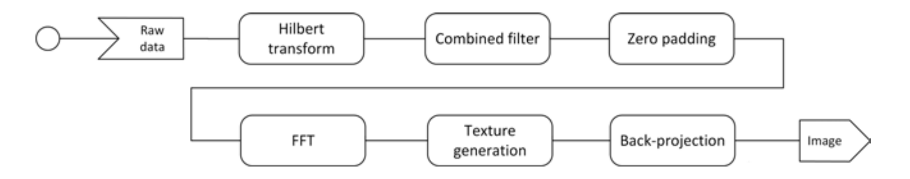


Figure 1: Activity diagram showing the steps of the radar imaging algorithm.

1 Sensor system

Based on recent research on MIMO imaging radar sensors and prior projects to measure the width of steel slabs in rolling mills [1] [2] [3] [4], a new sensor system is planned to not only measure the metal slab dimensions, but also to reconstruct high-resolution images of the material surface and to determine its speed.

The MIMO signal processing and its GPU implementation are based on a planned radar system with 197 real channels. The sensor operates according to the FMCW principle with a 3dB transmission range of 30GHz (119GHz − 149GHz). The sensor has a simulated resolution of 0.8mm along the vertical axis at a distance of 500mm. The combination of all transmit and receive channels provides a virtual array aperture of 1300mm distributed over 677 spatial positions with a virtual sampling distance of $2\lambda \approx 4.2$mm.

The transmitters are time-multiplexed, and only one transmitter is active at once. While the active transmitter is sending the chirp pulse, all other antennas act as receiver for the reflected radar signal.

2 Imaging algorithm

The input data for the algorithm is any number of pre-determined intermediate frequency (IF) signals $s_{\mathrm{TxRx}}(n)$ with size N from any transmitter (Tx) and receiver (Rx). The raw data is transmitted via ethernet using an UDP-based data protocol. Those given, the imaging algorithm consists of the steps shown in figure 1.

2.1 Signal pre-processing

The pre-processing involves a Hilbert transform function $\mathcal{H}$ to get the analytic input signal and the application of a complex combined filter function $\underline{\mathrm{w}}_{\mathrm{TxRx}}(n)$ with zero padding. In consideration of these two functions, the signal after the pre-processing step is given by (1).

$$\underline{\mathrm{s}}_{\mathrm{TxRx}}(n) = \begin{cases} \underline{\mathrm{w}}_{\mathrm{TxRx}}(n)[\mathrm{s}_{\mathrm{TxRx}}(n)+\mathrm{i}\mathcal{H}(\mathrm{s}_{\mathrm{TxRx}}(n))] & \text{if } n<N \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For the actual implementation, the analytic signal is approximated through a Fast Fourier transform (FFT) by setting all negative frequencies in the signal spectrum to zero followed by an inverse FFT [5]. The zero padding then appends a specified number of zeros to the filtered IF-signal to get a spectrum with lower peaks but higher distance resolution during the image reconstruction. With the speed of light c_0 and the radar bandwidth B, the enhanced step size of the range axis after the zero padding Δd is given by (2). The possible length N_P of the padded signal $\underline{\mathrm{s}}_{\mathrm{TxRx}}(n)$ will be determined through tests with the finished system.

$$\Delta d = \frac{c_0 N}{2BN_P} \tag{2}$$

2.2 Combined filter

The combined filter $\underline{\mathrm{w}}_{\mathrm{TxRx}}(n)$ in (3) consists of several sub-filters for different tasks. The calibration $\underline{\mathrm{w}}_{\mathrm{cal}}(n)$ removes channel response from the measurement signal, the Hamming filter $\mathrm{H}(n)$ and the Kaiser filter $\mathrm{K}_{\mathrm{TxRx}}$ suppress side lobes and noise in the imaging area and the multiplicity value $\mathrm{M}_{\mathrm{TxRx}}$ equalises the illumination level along the aperture of the MIMO array. The effect of the different filters is shown in figure 2.

$$\underline{\mathrm{w}}_{\mathrm{TxRx}}(n) = \underline{\mathrm{w}}_{\mathrm{cal}}(n)\frac{\mathrm{H}(n)\mathrm{K}_{\mathrm{TxRx}}}{\mathrm{M}_{\mathrm{TxRx}}} \tag{3}$$

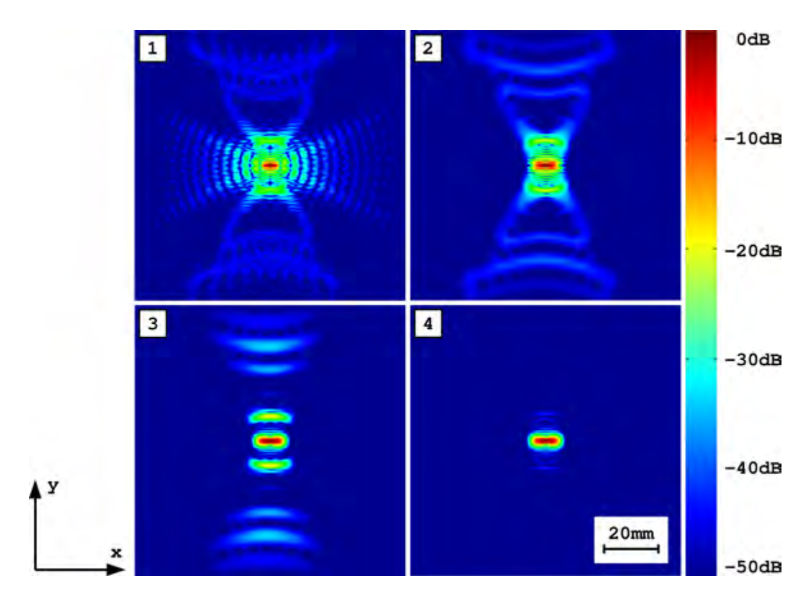


Figure 2: Output data $|\underline{I}(x, y)|$ of the imaging algorithm with 970 simulated input signals ($B = 160\text{GHz} - 120\text{GHz}, N = 1024, N_P = 8192$) for a single point reflector with (1) No filters (2) Hamming filter (3) Hamming and Kaiser filter (4) Hamming, Kaiser and Multiplicity filter.

The Hamming filter (4) is applied as a window function over the whole signal length in range direction reducing side lobes and noise along the x-axis of the reconstructed image.

$$\text{H}(n) = 0.54 - 0.47\cos\left(\frac{2\pi n}{N-1}\right) \tag{4}$$

The Kaiser filter is not applied over the signal length but over the y-axis of the antenna array. A virtual antenna position V_{TxRx} is calculated as the midpoint between the transmitter position P_{Tx} and receiver position P_{Rx}. The virtual y-position of each pair is then used to calculate the Kaiser value K_{TxRx} in (5). Here, I_0 is the zeroth-order modified Bessel function of the first kind.

$$K_{TxRx} = \frac{I_0\left[\pi\alpha\sqrt{1-\left(2\frac{V_{TxRx,Y}-\min(V_Y)}{\max(V_Y)-\min(V_Y)}-1\right)^2}\right]}{I_0(\pi\alpha)} \quad \text{with} \tag{5}$$
$$\alpha = 4.0$$

The multiplicity value M_{TxRx} for each antenna pair is generated by counting the number of overlapping virtual antenna positions within a threshold radius around V_{TxRx}.

2.3 Image reconstruction

The image reconstruction steps use a back-projection algorithm to map the filtered input signals onto a two-dimensional plane. Given by the physical properties of the FMCW radar, peaks in the IF-signal spectrum correspond to the presence of a reflecting object in the sensors' field-of-view. Therefore, the first step is to perform the FFT of the IF-signal. In (6) the signal response of an antenna pair $\underline{S}_{TxRx}(x, y)$ is calculated for any position in the target area with the distance $d_{TxRx}(x, y)$ between the antennas and the pixel position.

$$\underline{S}_{TxRx}(x, y) = \mathcal{FFT}\left[\underline{s}_{TxRx}(n)\right]\left(\frac{d_{TxRx}(x,y)}{\Delta d}\right) \tag{6}$$

Since the FFT is a discrete function, it is not possible to calculate the value of $\underline{S}_{TxRx}(x, y)$ directly. Therefore, it is necessary to interpolate the result of the FFT at the target distance to get a continuous function. The efficient implementation of this interpolation through GPU texture memory is one of the main aspects of the optimisation process described in this work.

Finally, the signal response of each antenna pair is superimposed to get the combined reflection intensity for any target position. With the application of a phase correction value $\underline{\Phi}(d)$, the reflectivity function $\underline{I}(x, y)$ can be represented as in (7). Here, f_0 refers to the starting frequency of the FMCW radar chirp.

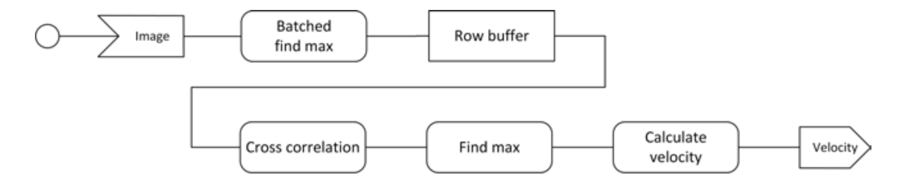


Figure 3: Activity diagram showing the steps of the velocity estimation algorithm.

$$\begin{aligned}\underline{\mathrm{I}}(x,y) &= \sum_{\mathrm{TxRx}} \underline{\Phi}\left[\mathrm{d}_{\mathrm{TxRx}}(x,y)\right] \underline{\mathrm{S}}_{\mathrm{TxRx}}(x,y) \quad \text{with} \\ \underline{\Phi}(d) &= e^{-\mathrm{i}\frac{2\pi f_0}{c_0} d}\end{aligned} \tag{7}$$

3 Velocity estimation

The velocity estimation is based on the imaging algorithm. With a MIMO array arranged along the movement axis of the observed object, the shift of the object is determined through the cross-correlation of two consecutive measurements. The steps of the velocity estimation in figure 3 involve the reduction of the image data to an one-dimensional function (8), from which the shift of an object is determined through a cross-correlation with the previous image data (9).

$$\mathrm{I}_{\max}(y) = \max_{x \in D_x}(|\underline{\mathrm{I}}(x,y)|) \qquad \text{with } D_x = [x_{\min}, x_{\max}] \tag{8}$$

The cross-correlation is calculated between each $\mathrm{I}_{\mathrm{max,i}}$ and the previous $\mathrm{I}_{\mathrm{max,i-1}}$ to determine the shift of the observed object. Here, $\star$ is the short notation for the cross-correlation.

$$\Delta y = \underset{y \in D_y}{\operatorname{argmax}}(\mathrm{I}_{\mathrm{max,i}}(y) \star \mathrm{I}_{\mathrm{max,i-1}}(y)) \text{ with } D_y = [y_{min}, y_{max}] \tag{9}$$

Finally, the velocity is calculated with the timestamps t_i and t_{i-1} of the received data (10).

$$v = \frac{\Delta y}{t_i - t_{i-1}} \tag{10}$$

4 Implementation details

The approach of this project is not only to implement the new algorithm in an efficient way, but also to ensure the re-usability of the developed code by creating the foundation for a general-purpose CUDA signal processing library. With that in mind, the focus during the development is on modularity, the creation of clean and safe code and a proper documentation.

The software is written in C++17 and is completely object-oriented. The different parts involve an advanced memory management, data handling, error handling and a flexible structure for the implementation of new arithmetic operations. Since CUDA code uses global definitions for the kernel and device functions and in some cases even needs global variables, all those relics from CUDA C are hidden behind proxy classes and never exposed to the user of the library. In order *to make error handling systematic, robust, and non-repetitive* [6, E.2], this library replaces the return-based error handling from the CUDA Runtime API with a throw-based error handling with dedicated exception classes.

4.1 Memory organisation

The CUDA Runtime API provides C-like `cudaMalloc` and `cudaFree` functions for memory allocation and deallocation. This technique is still supported but outdated in modern C++ [6, R.10] and therefore, those functions are wrapped in the class `DeviceArray` to perform an automatic allocation and deallocation in its Constructor and Destructor. Avoiding manual memory allocation reduces the risk of leaks and simplifies the memory management.

Another abstraction layer shown in figure 4 is the implementation of different `DeviceData` subclasses. Those subclasses hold additional information about the data dimension and provide methods to access subareas of the allocated memory without manual pointer manipulation.

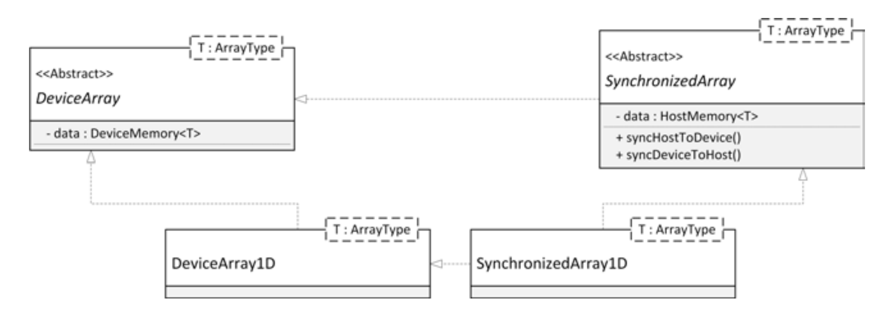


Figure 4: Simplified class diagram of the implemented memory management. Classes derived from `DeviceArray` inherit the automatic device memory allocation. Classes derived from `SynchronizedArray` inherit the automatic host memory allocation and synchronisation methods.

In addition, the `SynchronizedDeviceArray` classes allocate host memory and allow the bi-directional synchronisation of host and device memory. The `cudaMallocHost` method is used to enable a faster data transfer during the synchronisation. Virtual inheritance is used to resolve the diamond pattern in this design.

All memory management classes are templates to be usable with different data types without code duplication. Those templates are specialised through explicit template specialisation, which generates the source code for a specific selection of data types during compile time. This technique is mainly used to restrict the usage of the template classes to only implemented and tested data types. This explicit form of generating source code from templates during compilation (template metaprogramming) is not recommended in the C++ Core Guidelines [6, T.120,T.121], except for the emulation of concepts. Although this library would effectively benefit from using concepts, the Nvidia Nvcc compiler does not support this new C++20 feature yet.

4.2 Arithmetic operations

All arithmetic functions and various memory operations are implemented using a visitor pattern. The design shown in figure 5 implements the different operations as visitors in dedicated classes which get called by the memory objects. This design avoids hard binding

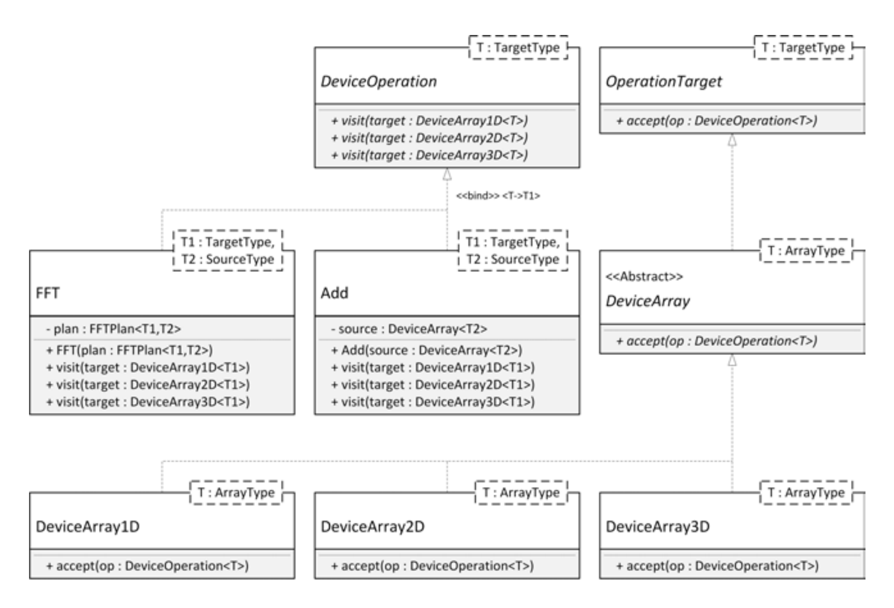


Figure 5: Simplified class diagram of the implemented visitor pattern for the arithmetic operations. Any `DeviceOperation` can be applied on any `OperationTarget`. The two operations `FFT` and `Add` are examples of concrete `DeviceOperation` classes.

between the implemented operations and the data on which those are applied and facilitates the implementation of new operations.

An alternative to this design would be a Utility Class implemented as Singleton or with the use of static methods. There is a wide discussion about the usage of Utility Classes, and in general they are not seen as good practice. They break the principles of object-oriented programming by having only one instance of the class, which comes with several downsides compared to a regular instantiable class. Moreover, the tight coupling to the Utility Class prevents to switch this dependency by creating a subclass and extending its functionality.

4.3 Optimisations

The final implementation of the image processing algorithm is the result of a longer optimisation process, and a selection of the interme-

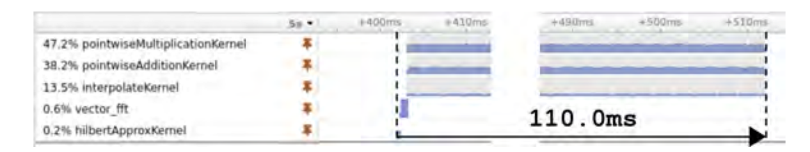


Figure 6: Nvidia Nsight Systems timeline for the unoptimised algorithm. Many small kernels are launched and executed sequentially.

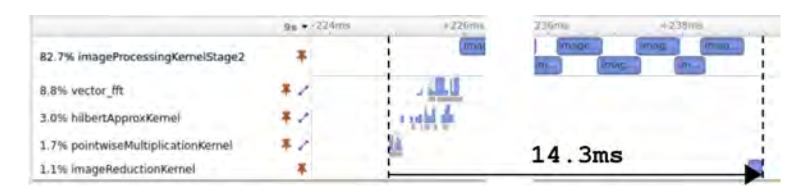


Figure 7: Nvidia Nsight Systems timeline for the second optimisation stage. A specialised image processing kernel per data block massively reduces the overhead through many small kernel launches.

diate stages of this process is presented and compared in this section. Those stages of the development process are described to explain the different design decisions and their effect on the overall performance. For comparison, all described tests are performed on the same hardware with 970 simulated input signals ($N = 1024, N_P = 8192$) and a target area size of 512×512 pixel. The execution time is measured with the analysis software Nvidia Nsight Systems, and the GPU activity is observed with Nvidia Nsight Compute.

The first attempt to implement the image processing does not involve any specialised kernel functions but only uses general arithmetic operations. This stage exclusively aims to check the general functionality and to determine the upper bound of performance improvement. All operations in figure 6 are executed sequentially with a total execution time of 110.0ms.

As a first optimisation step, the input data is divided into different blocks to enable the parallel execution of multiple kernel functions on the GPU. Each processing block generates partial image data, which is combined to the final image when all processing blocks are finished. The assigment of a dedicated CPU thread and CUDA stream to each block reduces the overall execution time to 94.3ms.

The most obvious drawback of the previous implementation is the high amount of small kernel launches, which comes with a large overhead compared to a single specialised kernel. The calculation of the antenna distance, the interpolation of the FFT data and the application of the complex phase correction consists of five kernel launches for each signal. The idea is now to combine those steps in one kernel launch per block and to loop over the signals inside the kernel function. The result in figure 7 is a reduction of the overall execution time down to 14.3ms.

The most significant proportion of the execution time is still caused by the specialised image processing kernel, and reducing its execution time will have a large impact on the overall execution time. Further optimisations and the analysis of a single kernel need a deeper look into the GPU hardware. The kernel analysis tool Nvidia Nsight Compute measures various metrics of a kernel on the hardware layer. That includes bandwidth measurements, the utilisation of the different memory types, cache and hit-rate analysis and the utilisation of the different GPU pipelines.

The analysis of the image processing kernel reveals a very low L2 cache hit-rate of 5.66% and a very high utilisation of the GPU integer multiplication and floating point operation pipeline (FMA). Both problems are targeted by transferring the FFT data into a batched read-only 1D-texture in which each row is filled with the complex spectrum of a single input signal. Thus, the GPU is able to perform more aggressive caching by ignoring possible write operations and predicting the memory access for adjacent rows. Another huge advantage of the texture memory is the ability to perform a hardware interpolation during memory access which relieves the FMA unit. The analysis of the optimised kernel shows a L1 and L2 cache hit-rate of over 95%, a balanced utilisation of the used GPU pipelines and an overall execution time of 6.1ms.

5 Conclusion

The new imaging radar algorithm was successfully implemented in CUDA C++ and was used to perform initial simulations and to estimate the expected signal processing time of the sensor system. Besides

general optimisations, like the usage of streams and the step-by-step kernel runtime optimisation, the main outcome is the suitability of the GPU texture memory for the back-projection algorithm. A first sensor prototype is in the making and will be used for further tests and to verify the results of this work. The code developed during this work is now the foundation for a general-purpose CUDA signal processing library, which will be used and extended in further projects.

References

1. C. Schwäbig, S. Wang, and S. Gütgemann, "Development of a millimetre wave based sar real-time imaging system for three-dimensional non-destructive testing," *tm - Technisches Messen*, vol. 88, no. 7-8, pp. 488–497, 2021.
2. J. Romstadt, H. Papurcu, A. Zaben, S. Hansen, K. Aufinger, and N. Pohl, "Comparison on spectral purity of two sige d-band frequency octuplers in mimo radar mmics," in *2021 IEEE BiCMOS and Compound Semiconductor Integrated Circuits and Technology Symposium (BCICTS)*, 2021, pp. 1–4.
3. E. Tolin, M. A. Campo, H. Cetinkaya, R. Herschel, S. Gütgemann, C. Krebs, and S. Bruni, "Uwb millimeter-wave 1d mimo array for non-destructive testing," in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, 2021, pp. 1–5.
4. M. Ortner, Z. Tong, and T. Ostermann, "A millimeter-wave wide-band transition from a differential microstrip to a rectangular waveguide for 60 ghz applications," in *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*, 2011, pp. 1946–1949.
5. L. Marple, "Computing the discrete-time "analytic" signal via fft," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2600–2603, 1999.
6. B. Stroustrup and H. Sutter. (2021) C++ core guidelines. Last visited 29.09.2022 15:48. [Online]. Available: https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines

Deep learning and active learning based semantic segmentation of 3D CT data

Markus Michen[1] and Ulf Haßler[1]

Fraunhofer-Entwicklungszentrum Röntgentechnik EZRT,
Flugplatzstr. 75, 90768 Fürth

Abstract In this paper, we developed a tool that uses active learning and deep learning together for segmentation of 3D CT data. We demonstrate the results of the method using the use case of plant segmentation. In addition, we compare the method with a baseline and a classical image processing-based algorithm.

Keywords Deep learning, active learning, semantic segmentation, plant segmentation, image processing, u-net

1 Introduction

Automated segmentation of 3D CT data is a vast field of application. Especially in the medical environment, there is currently a transition from conventional methods based on classical image processing to Machine Learning / Deep Learning (ML/DL) based methods [1,2]. Much of the aforementioned success of Deep Learning is due to the large number of publicly available annotated datasets, for example, the ImageNet database [3]. One of the major challenges is the necessity to acquire sufficient ground truth data for modeling. However, this data are usually not available in sufficient quantities, especially for industrial use cases. Moreover, the annotation of this data turns out to be an extremely time-consuming and very expensive task, especially for large 3D datasets.

Thus, we need effective methods to reduce the labeling effort. One such method is active learning, a collection of techniques that support machine learning algorithms to achieve better results with less labeled

training data. The learning algorithm can interactively prompt a user to assign the correct labels to new data points. To do this, the algorithm should ask questions that promise a high information gain in order to keep the number of questions as small as possible.

These questions, called queries, can be grouped into three main types: stream-based selective sampling, membership query synthesis, and pool-based sampling. Stream-based selective sampling assumes a stream of incoming unlabeled data points x. The current model and a measure of informativeness measure $I(x)$ are used to decide for each incoming data point whether to ask the oracle for an annotation. In membership query synthesis, the data points are not drawn, but rather the model generates new data points in a way that it considers informative to itself. With pool-based sampling, a batch b is selected from the unlabeled dataset. The current model is used to predict the sample stack and obtain a measure of informativeness $I(b)$. Based on this measure, the best N samples are selected to be annotated by the oracle [4].

Overall, Deep Learning has strong capabilities in processing data through automatic feature extraction, but requires a very large amount of annotated data to do so. Active Learning, on the other hand, has the potential to effectively reduce the effort required for labeling. The combination of deep learning and active learning support each other, so their application potential improves significantly. Therefore, we have developed a tool that allows us to apply active learning to the area deep learning segmentation of 3D CT data.

We demonstrate the use of our tooling on the basis of plant segmentation, as plant breeding has undergone rapid progress in recent decades. In this context, targeted plant breeding, for example of climate-resistant strains, is also becoming increasingly important [5]. Innovative analysis methods, such as 3D segmentation, play an essential role in this context, enabling seedlings and seeds to be assessed qualitatively.

The segmentation task here is to divide the 3D CT scan of the plant inside a container in folded paper into the classes plant, paper and background (see figure 1). Through use of the segmentation the individual plants can be evaluated and classified by downstream applications later. It is particularly difficult to distinguish the seedlings from the paper. Paper and seedling absorb X-rays to a similar degree, so there is virtually no usable contrast difference that could be used for

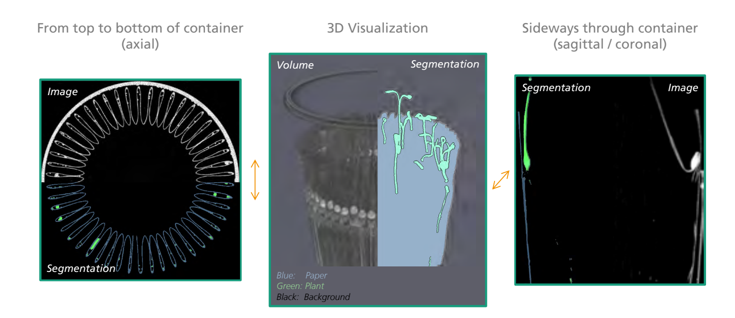


Figure 1: Data for the plant segmentation. The data is split in raw image or volume and its corresponding segmentation. On the left the volume is sliced in axial direction. In the middle a 3D rendering can be seen. And on the right the sagittal/coronal direction is shown.

segmentation. This is also affected by the limited resolution of only 140 µm and noise, which is why incorrect segmentations can easily occur. Either components of the seedling are assigned to the paper or vice versa. This hinders the subsequent assessment of the seedling in the downstream application due to incorrectly calculated characteristics.

2 Methods

Our method operates in three main phases (see figure 2). In the pre-training phase, an initial network (currently 3D U-Net) is trained from weak labels. These can be derived from existing classical image processing pipelines, simulations or rough hand-annotations.

Subsequently, this pre-trained network is passed to the active learning phase. The active learning phase itself also consists of several steps, namely inference, location, visualization/interaction, and training. During inference, the segmentation network generates a segmentation map, which is then analyzed during the location phase. Then the user can visualize the results and interact with them to correct invalid segmentations. Next, the areas corrected by the user are retrained during the training phase and the weights of the segmentation network are updated. A graphical user interface guides the user through these four steps until a visually satisfactory result is achieved or an application-specific condition is met.

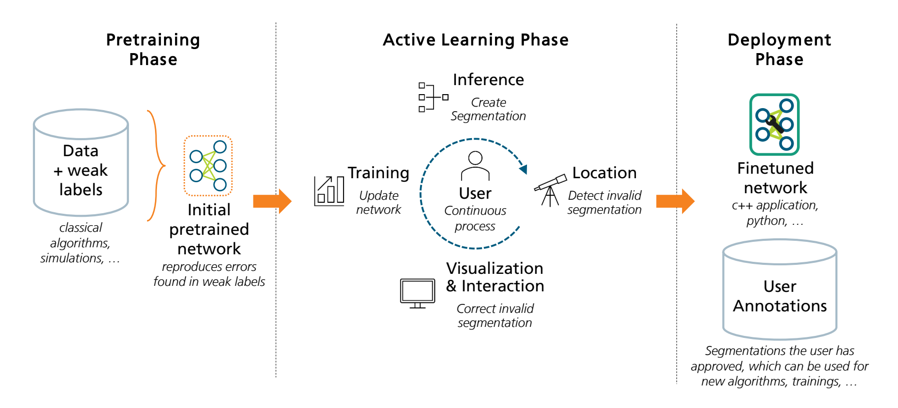


Figure 2: Overall conceptual process of the developed deep learning and active learning approach.

Finally, the resulting fine-tuned network can be deployed. As an additional result, all corrections made by the user during the active learning phase can be used for future algorithms or training.

2.1 Pre-training phase and network architecture

In the pre-training phase, the network is initially trained in such a way that later, in the active learning phase, the segmentation is almost correct and only invalid segmentations have to be corrected and re-trained. For this, already existing classical algorithms (based on thresholding, filtering, . . .) or simulations can be used as weak labels.

The U-Net architecture used consists of a simple 3D U-Net (see figure 3). It is 5 levels deep with 2 convolution blocks per level. With each level the number of feature maps doubles and the spatial resolution halves. The convolution blocks consist of 2 convolution layers with batch normalization [6], swish activation [7] and a residual connection. In the decoding path the feature maps are upsampled and concatenated by simple upsampling. The last layer is $1x1x1$ convolution with Softmax activation and represents our final segmentation. The entire 3D input volume is usually too large to be processed at once, so it is processed block by block through subvolumes of size 64^3.

The training parameters are set as follows. As loss function we

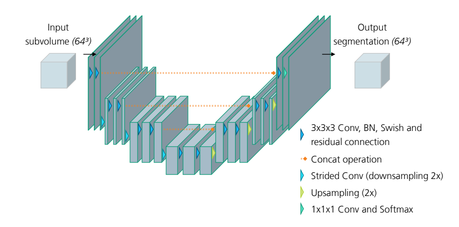


Figure 3: Schematic of the employed u-net network architecture.

choose the sum of dice and cross entropy loss, as in [8]. For the optimizer, we use Adam [9] with a learning rate of $3e^{-4}$ and cosine decay. As regularization, we set a weight decay of $2e^{-5}$ [10] and also use label smoothing of 0.1. Additionally, we use augmentations to increase the training data. We use contrast, noise, affine transformations, flips, blur and artifacts augmentations with varying strength. The network is implemented using TensorFlow [11] and the augmentation pipeline makes use of the TorchIO package [12]. The training has been conducted using a NVIDIA GTX Titan X with 12 GB of GPU RAM.

2.2 Active learning phase

After the network has completed the pre-training phase and has reached suitable convergence, it is passed on to the active learning phase. Here the user is in the focus, and first he is presented with the following view within the simple application we developed, with which he can interact with the current dataset. In figure 4 three orthogonal sliceable views with which the dataset can be navigated can be seen and the toolarea in which multiple tools are available for the user. The user has access to brush, image processing operations (flood-fill, morphology, clustering, ...), 3D visualization, neural network training and use-case specific functions.

In general, 4 sub-steps are then performed within the active learning phase: namely, inference, location, visualization/interaction, and training.

Inference. Here the network prediction is executed. In the field of 3D CT data segmentation, the volumes are often large, with sizes of several GB or more, which prevents the direct use of a segmentation network

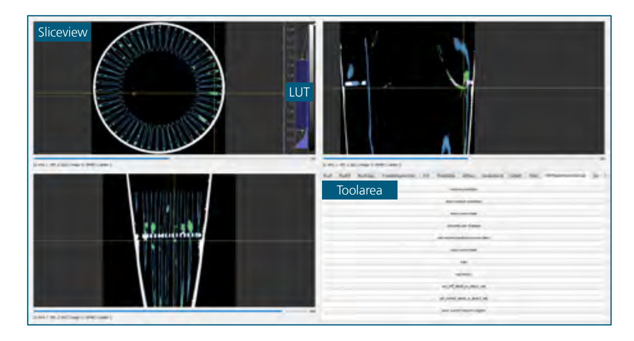


Figure 4: Overview of the graphical user interface guides the user through the active learning phase. In the top right, top left and lower left, orthogonal sliceable views can be seen, that allow the user to navigate the through the data and overlayed segmentation. In the bottom right the toolarea can be seen.

due to the limited GPU RAM. Therefore, to perform segmentation with such volumes, we need to split them into smaller blocks (usually 64^3). Each of these blocks is then segmented individually by the network. In addition, overlapping is performed at the edges of the blocks to compensate for the lack of spatial information at the edges. Finally, all the blocks are merged to form the total volume.

Location. In the localization phase, the user has to find and correct incorrect segmentations. Since this is a very time-consuming process to do manually, we have developed a way to quickly and semi-automatically present potentially incorrect areas to the user. To do this, we use a random forest that classifies the objects contained in the current segmentation. It is trained by the user on the basis of a few examples. For this purpose, first the current segmentation is analyzed by a connected component analysis (CCA). Then, features are calculated for each of the connected objects (e.g. size, mean, eigenvalues, ...). Now the user has to label at least one object of each desired class (for example: paper, seedling, faulty, ...). After that, the random forest can be trained and applied to all contained objects. The user is then shown the objects that have been classified and can then improve their segmentation. The GUI and the random forest pipeline are shown in figure 5.

Visualization/Interaction. After a wrong segmentation has been

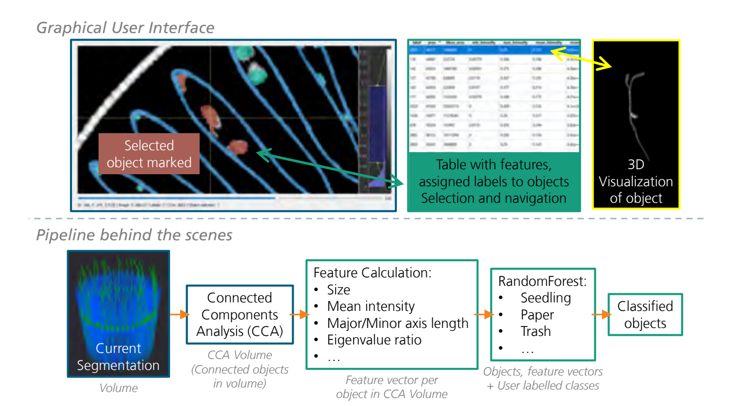


Figure 5: Location phase overview. On the top the GUI part of the location phase can be seen. In the middle the table containing features of the objects can be selected. On the left you can see a cutout of the data and the object is slightly highlighted in red. On the right a 3D visualization of the object is displayed. The lower part shows the pipeline running in the background. The gray text describes the data flow from plain voxels to connected objects and their features.

found in the localization phase, the user has to correct it. This is done with the help of the three orthogonal views in the GUI and the available tools. Most of the time, the corrections that need to be made are small local corrections, such as roots that are incorrectly marked as paper. However, painting pixels is difficult and painting voxels turns out to be even more difficult. That's why we provided the brush tool with special modes for segmenting plants. After all, the brush tool is the most commonly used tool for local segmentation changes. It should be easy (and fun) to use and support many automatic modes so that the user can segment as many voxels as possible by hand with as little effort as possible. In figure 6 the brush usage of the brush tool is shown along with its special Frangi [13] filter mode.

Training. After the localization, visualization and interaction phases mentioned above, the training phase can begin. The goal of our active learning process is that the user annotates as little as possible, but as much as necessary to correct the wrong segmentations. Therefore, the changes in the loss of the network are also rather small, which could hinder the learning of the corrected regions. To compensate for this

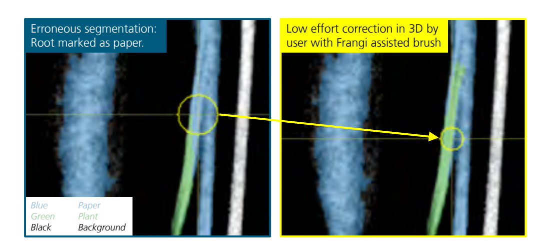


Figure 6: Example of an incorrect segmentation and its correction. The Frangi filter can select tubular structures, which makes it easier to separate them from the planar paper, allowing the user to correct the incorrect segmentation more easily than by manually tracing each voxel.

imbalance, voxel-wise loss weighting is used to force learning of the regions corrected by the user. The weight calculation is similar to scikit-learns class weights function [14]. The training parameters are the same as in the pretraining phase mentioned above.

Iteration. Finally, the figure 7 shows an iteration of the active learning process of the developed tool. Starting in the inference phase, the current network generates a segmentation. Then, in the localization phase, the incorrect region is found and presented to the user. Subsequently, the user corrects the incorrectly segmented voxels. After finishing training with the new annotations, the next iteration can start. In the upper right of the figure 7, the result after the iteration can be seen on another area that was not annotated by the user.

2.3 Deployment phase

After the active learning phase has been completed, the resulting fine-tuned network can be passed on to the deployment phase. Here, it is then used for inference in another application. In the case of plant segmentation, the output of the network is used to analyze individual seedlings and their characteristics for subsequent seed selection and breeding.

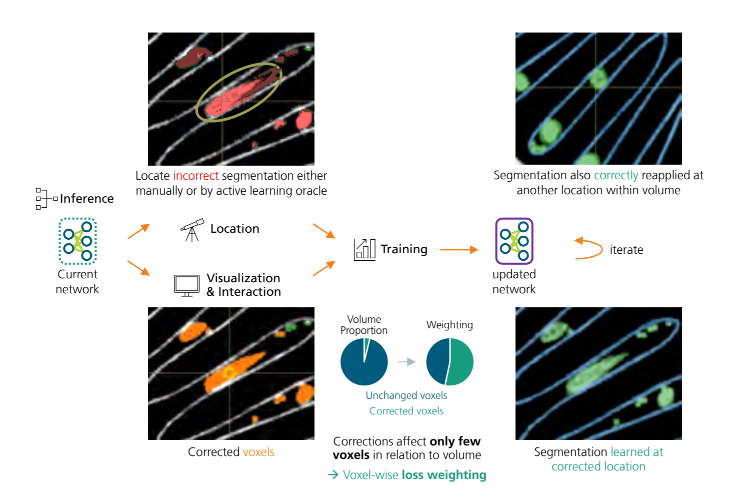


Figure 7: Overview of the usual active learning workflow with example segmentations in the different steps. In the top right the result after the iteration is shown.

3 Results

We evaluated our developed tool on the use case of segmentation of 3D CT scans of plants. The seedlings grow in a plastic container in folded paper. Due to the similar attenuation coefficients, it is particularly difficult to distinguish plant and paper. We compare the performance of the pre-trained network and the fine-tuned network with the performance of a classical image processing-based algorithm [5]. The methods are compared visually by inspection and by calculating segmentation metrics. To give no algorithm an advantage, we manually created a ground truth scan from the test set from scratch without using algorithmic assistance. In order not to let the effort explode, we evenly distributed two slices from each of the three orthogonal directions (see figure 1) for annotation. Each of these six slices took the annotator an average of 20 minutes, extrapolating to the total scan size of about 800^3, this would require about 16 days for the entire scan in the worst case, which would be impractical.

Figure 8 show the comparison of the segmentation with the respective ground truth slice from the two different directions. As can

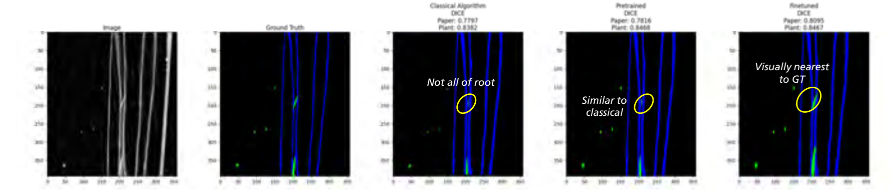


Figure 8: Results for the ground truth slices of different directions (top: axial, bottom: sagittal/coronal). The method names and their dice results are shown above the image. Various points of interest are highlighted by yellow ellipses.

be seen, the classical algorithm generally segments the roots more sparsely than the ground truth. In some cases, the roots are completely missed, which is a fatal error for the final application in the deployment phase. The pre-trained network reproduces the errors of the classical algorithm, which is to be expected after it has been trained with data from the classical algorithm. The fine-tuned network finds roots missed by the other two methods, but segments them a bit too thick. Nevertheless, such an error is not as serious as missing roots.

The figure 9 shows the metrics of the different methods. It can be seen that all metrics are quite close to each other. The classical algorithm can only convince in one metric, while the pre-trained network achieves the highest score in 2 out of 12 cases. In the remaining 9 out of 12 cases, the fine-tuned network achieves the highest scores. This is also in agreement with the assessment in the visual inspection.

4 Conclusion

Overall, the results achieved with our active learning tooling in plant segmentation are very promising. Although all metrics are quite close to each other, we have a performance gain of about 1% in terms of the DICE score. Furthermore, qualitatively visually, the DL segmentation

Metric	Accuracy			Area under curve			Dice			Intersection over Union (IoU)		
Algorithm	Background	Paper	Plant	Background	Paper	Plant	Background	Paper	Plant	Background	Paper	Plant
1_classical	0.9827	0.9825	**0.9962**	0.9516	0.9495	0.8499	0.9906	0.8708	0.7456	0.9815	0.7735	0.6064
2_pretrained	0.9825	0.9827	0.9958	0.9630	0.9572	0.9019	0.9905	0.8741	**0.7583**	0.9812	0.7789	**0.6178**
3_finetuned	**0.9837**	**0.9848**	0.9949	**0.9788**	**0.9682**	**0.9636**	**0.9911**	**0.8894**	0.7545	**0.9824**	**0.8031**	0.6109

Figure 9: Table of the calculated segmentation metrics. In the top row, the metric can be read. In the second row, the class to which it refers. The last three rows show the results of the individual algorithms. The metric of the best method is highlighted in green.

results are ahead. Additionally, we did not use any prior knowledge about scan geometry, container, paper and plant type. This makes the DL approach much easier to adapt. In the future, other active learning approaches or new network architectures can be integrated to further increase the performance.

Acknowledgments

This work was supported by the Bavarian Ministry for Economic Affairs, Infrastructure, Transport and Technology through the Center for Analytics Data Applications (ADA-Center) within the framework of "BAYERN DIGITAL II".

References

1. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
2. Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.
3. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
4. B. Settles, "Active learning literature survey," 2009.

5. M. Rehak, U. Haßler, T. Grulich, N. Wörlein, F. Porsch, I. Götz, and A. Wolff, "Der phenotest–ein automatisiertes ct-system zur phänotypisierung von zuckerrübenkeimlingen," in *Forum Bildverarbeitung 2018*. KIT Scientific Publishing, 2018, pp. 217–228.
6. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.
7. S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.
8. F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert *et al.*, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.
9. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
10. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
11. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/
12. F. Pérez-García, R. Sparks, and S. Ourselin, "Torchio: a python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning," *Computer Methods and Programs in Biomedicine*, p. 106236, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260721003102
13. A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *International conference on medical image computing and computer-assisted intervention*. Springer, 1998, pp. 130–137.
14. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn:

Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Signal processing pipeline for an autonomous electrical race car

Martina Scheffler[1], Ole Kettern[1], Oliver Zbaranski[1,2], Finn Schäfer[1,2], Kevin Schmidt[1], Bjarne Eberhardt[1], and Stefan Werling[2]

[1] CURE Mannheim, Autonomous System Team, Handelsstraße 13, 69214 Eppelheim
[2] Duale Hochschule Baden-Württemberg Mannheim, Coblitzallee 1-9, 68163 Mannheim

Abstract This work presents a signal processing pipeline for an autonomous race car in the context of Formula Student. The software used for each step from the detection of objects in camera images or lidar point clouds to the calculation of control outputs for the actuators is described in detail. The sensors and actuators are covered and the system output is visualized. The computational times of the pipeline are analyzed and it is derived that the complex algorithms used for motion planning and SLAM take up the most of the computation times, leaving the most room for improvements.

Keywords Autonomous driving, signal processing, Formula Student, YOLO, object detection, SLAM, MPC

1 Motivation

The future lies in autonomous driving, at least in the Formula Student (FS), an international design competition between student teams. In this work, the signal processing pipeline of the 2022 electrical and autonomous race car of the team CURE (Cooperative University Racecar Engineering) is presented. While the Formula Student poses a rather narrow challenge for autonomous vehicles due to a controlled environment and clearly specified tracks and track boundaries, it is a good development and testing ground for algorithms which are also used in agricultural, industrial or real-life traffic situations.

2 Problem description

During the FS events, the autonomous race car competes in four different types of competitions, all posing different challenges to the car and the Autonomous System (AS): Acceleration (1), Skidpad (2), Autocross (3) and Trackdrive (4). The disciplines test the car's ability to (i) drive straight lines (1), (ii) handle high acceleration and deceleration forces (1), (iii) withstand high lateral forces (2), (iv) choose the correct direction at a known intersection (2), (v) navigate unknown tracks (3) and (vi) reliably generate global maps and locate itself in them (3, 4). During all events, the track boundaries are marked by cones of known sizes [1, Tab. 3]. Small blue and yellow cones mark the left and right sides and orange cones signal finish lines and the exit areas in which the car has to come to a standstill. The challenge the cars face is to detect the cones correctly, align the detections with previous knowledge about the tracks - either from the competition rules or from internally built maps - generate a path to follow and send control signals to the actuators accordingly.

3 System overview

This section gives a brief overview of the hardware and software used to run the pipeline. In it, the processing unit, the sensors and the actuators of the race car are described.

To handle the challenges regarding the computational power and the needs of the image processing software, a custom-built Autonomous Compute Unit (ACU) consisting of an AMD Ryzen 5 5600G hexa-core CPU, a NVIDIA Tesla T4 data center GPU and 32GB of memory is used. On it, Ubuntu 20.04 LTS is installed. To implement the various functionalities of the AS, the Robot Operating System (ROS) Noetic is used. This provides the means for inter-process communication, threading, debugging as well as visualization tools. In order to simplify development, deployment and maintenance, the complete AS is containerized using Docker.

To interact with the rest of the electrical system in the race car, multiple CAN buses are used. To send / receive messages, a CAN to ROS interface is used. The sensors connected to the CAN bus include steer-

ing wheel angle sensors, wheel speed sensors and an IMU (Inertial Measurement Unit). All of them are used as inputs to the AS in order to track the car's position and generate control outputs accordingly. These are then used to control the actuators which include the motors, the motor for the steering actuation and the electrical valves for the brake system. Additionally to the sensors connected via CAN, other sensors are directly connected to the ACU via either USB or Ethernet. These include a dual-antenna GPS for position and heading information, a stereo camera and a lidar.

4 Signal processing pipeline

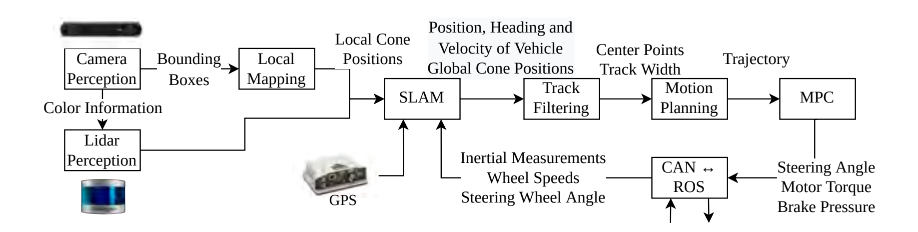


Figure 1: Overview of the modules of the signal processing pipeline.

This section gives a detailed description of the signal processing pipeline as a whole and each module in it as shown in Figure 1.

4.1 Camera perception

This section focuses on the generation of local maps from images taken with a Stereolabs ZED2i stereo camera.

Camera Calibration Since the camera images are currently used as the main way to determine the positions of the cones, the camera needs to be calibrated as precisely as possible. The local mapping process is closely related to this as it requires both an intrinsic and an extrinsic camera matrix to describe the transformation of the cone positions from image to world coordinates. With the currently used camera model,

the intrinsic matrix is supplied by the manufacturer and not subject to change. As of right now, the extrinsic calibration process is done by mapping image points to world points. In this case, our world points are represented in a 3×3 marker pattern whose position we obtain by measuring the distance to the camera itself. After capturing an image, coordinates of the markers in the image are collected by picking out the respective pixels. Using the image points, the intrinsic matrix and the world-coordinates of the points, the extrinsic matrix can be computed using OpenCV's function *solvePnP()* [2]. This method has the benefit of only using one image but the measurements of the world coordinates by hand and the determination of image points are error-prone and add a certain error to the calibration as a whole. Replacing the manual steps by automated library functions would bring a huge improvement to the accuracy of the resulting calibration.

Inference An integral part of the camera-based perception is the detection of differently colored cones in the images the camera provides. As these detections are used to calculate the position of the cones relative to the vehicle, the task of inference needs to be done both quickly and accurately.

In order to reach this goal, a neural-network-based approach for object detection was chosen. The core element is a YOLOv5 convolutional neural network [3], completely based on PyTorch, which makes it easier to work with. YOLO networks gained a lot of popularity in the last years as they achieve similar, if not better, accuracy than Single-Shot Detectors while being significantly faster [4]. Using the repository code, a network is trained using both images that were captured and labeled by ourselves, as well as additional training data from the Formula Student Objects in Context (FSOCO) repository [5]. To further improve the process, pre-trained weights are used which reduces the need for a big data set and, consequently, also the time needed for training.

The actual logic for the task of detecting cones is based on an open-source inference implementation of YOLOv5 that leverages the capabilities of NVIDIAs TensorRT library to further optimize performance [6]. Using this camera-based perception pipeline, the vehicle is able to detect cones in a distance of up to 15 meters on images with a resolution

of just 1280 × 720 pixels while achieving inference speeds of around 50ms on average. While the detections proved reliable under varying weather conditions, the neural network struggles in detecting cones when there is a strong backlight present, as the camera automatically lowers the image brightness as a consequence. Additionally, because of the the Python implementation, the processing of 1920 × 1080 pixel images that are provided by the camera is not possible without significantly sacrificing inference speed. Consequently, the migration of the code to C++ would be beneficial in the future.

Local Mapping In order to generate a map with reference to the car's current position, a translation of cone positions from image to world coordinates is necessary. The intrinsic and extrinsic matrices are used to project the top middle point of the bounding boxes around the detected cones from image to world. This projection results in a ray as the distance can not be calculated with only the pixel coordinates. To get the accurate position, the ray is intersected with a plane at the known height of the cones. This is done for all bounding boxes in the image resulting in a list of local cone positions to pass on to the rest of the pipeline.

While the calculation of the local maps itself has proven reliable during testing and competitions, its accuracy is highly dependent on the accuracy of the camera calibration, so an improved calibration process as mentioned above could significantly improve the quality of the local maps.

4.2 Lidar perception

To increase the robustness of the system as a whole, a Velodyne VLP-16 Puck Hi-Res lidar is used to generate local maps of the environment as well. For reasons of time, the lidar perception module has not actually been used during this year's competitions, However, development and tests with a test data set have been done.

First, the amount of data in the captured point cloud is reduced significantly by cropping the field of view in order to increase the computational performance. Second, the ground plane is filtered out using the Himmelsbach algorithm [7]. Once the point cloud only contains

points which are not in the ground plane, Euclidean Clustering is used to group the points. Then, the shapes of these clusters are checked to keep any cone-shaped clusters and remove erroneous detections like people, walls and other structures. The coordinates of the detected cones are then passed on to the SLAM and track filtering modules. Additionally, 3D to image translation is used to add color information to the detected cones using information from the camera images. As a side note, it has to be added that while the approach works well, the performance of the pipeline is limited by the low number of channels of the Puck Hi-Res. Cones which are 6m away from the lidar already consist of less than 10 points and the number of points decreases further with increasing distance.

4.3 Simultaneous localization and mapping

The main goal of Simultaneous Localization and Mapping (SLAM) is enabling motion planning to generate global trajectories and thus, increase the vehicle performance. The SLAM algorithm is implemented as an Unscented Kalman Filter (UKF) in Python. This type of filter was chosen as it is able to handle highly non-linear problems like polar cone positions more sufficiently than an Extended Kalman Filter (EKF). Also, it outperforms Particle Filters or Graph-based SLAM approaches due to their higher complexity. The underlying architecture and mathematics are based on the open-source library *FilterPy* [8], however adapted to increase speed and compatibility to our system. The tracked state vector $\vec{x}$ of the UKF consists of the tracked landmarks x_1, y_1 to x_n, y_n and the vehicle pose containing the vehicle position x, y, longitudinal vehicle velocity v_x and global vehicle heading ψ:

During the prediction, the system propagates through a simplified bicycle model disregarding any lateral forces and slip angles. The current steering wheel angle is used to calculate the travelled distance of the current cornering, while the current longitudinal acceleration a_x and yaw rate $\dot{\psi}$ measured by the IMU are used to calculate the new vehicle velocity and global vehicle heading. To continuously update the values, the output of the local mapping and lidar perception as well as the measurements of all four wheel speed sensors and the GPS are used.

To counter the disadvantage of the $\mathcal{O}(n^3)$ complexity of the UKF

algorithm with n being the number of state variables, the predicted state variables are limited to the vehicle pose states and the updated state variables are limited to the necessary ones, for instance, only the vehicle pose if no lidar perception or local mapping output is available and otherwise the vehicle pose and the observed landmarks. As a result, the complexity is nearly constant since the number of observed landmarks is naturally limited.

4.4 Track filtering

The track filtering module calculates the center point line of the track and the track width using the position and color of the cones. The general functionality of this module is split up into local and global filtering, based on the information passed on by the SLAM algorithm.

The local track filtering follows three steps. First, it finds the midpoints of the track using different approaches based on the number and color of cones available from SLAM. For only cone or one color available either the Dynamic Window or Border Shift approach is used. If more cones of each color are passed on, then the midpoints are calculated with the Delaunay Triangulation. With the variety of possible approaches, the reliability of this module can be enhanced. The second step is to interpolate and approximate the center line from the found midpoints. The third and last step is the definition of the legal track width for each point and the calculation of the left and right borderline. This information is then passed on to the motion planning module.

The global track filtering works very similar to its local counterpart, except that it uses only the Delaunay Triangulation for finding the midpoints, since all global cone positions are known. They are sorted with a tree algorithm and used to calculate the track width and border lines.

4.5 Motion planning

The goal of the motion planning module is to generate a trajectory to enable dynamic racing maneuvers. Therefore, it is separated into two parts: local and global. The local one is used when no closed global track is passed on by the SLAM algorithm. It is also used while the global optimization is still calculating the optimal race line for the closed and global track.

Local Motion Planning The local motion planning uses a directed geometric graph-based approach fully written in Python and based on [9]. The current vehicle position is used as origin. In regards to the centerline of the track, normals are calculated at regular intervals. The layers of the graph are made up of nodes which are evenly spaced on the normals. From one node, an edge to every node on the next layer exists. To generate a curvature-optimized race line, a cost is calculated for each edge. The cost takes into account the average and maximum of the squared curvature of the edge and its length. Using the known costs of all edges, the cheapest path can be found. The least-cost path represents the most curvature-optimal path, for which a velocity profile is then calculated. This velocity profile is calculated based on the hypothesis that the lateral velocity of the car at the apex point of a curve is 0 $\frac{m}{s}$. Due to this hypothesis, the maximum accelerating and decelerating velocity profiles are calculated from a ggv-map - it delivers the maximal acceleration forces - these two profiles are then superimposed.

Global Motion Planning The global approach is also based on curvature optimization and inspired by [10]. For generating the global race line, the problem is set up as a quadratic programming problem. The global algorithm tries to minimize the sum of the curvature for a given reference line. In this specific use case it is the closed global center line that is passed by the SLAM algorithm and used as reference line. In the following the quadratic solver tries to minimize the curvature via moving the way points on their normal vectors. The output path is then shifted into a trajectory using the same velocity profile calculation as the local approach. The global approach uses more computing power and takes more time to be calculated. Therefore, as mentioned above, the local optimization algorithm continues until a global trajectory is determined.

4.6 Model predictive control

The control module uses the trajectory from the planning module and the vehicle states from SLAM as input to control the vehicle dynamics. More precisely, the goal is to control the vehicle movement along the planned path. This *Path Tracking* problem [11] aims to minimize

the delta between the vehicle and the path points as well as to assure progress along the race track. A nonlinear model predicitve controller (NMPC) was developed to reach this goal. In general, a NMPC consists of three key components: A nonlinear vehicle model, an optimization objective and a reliable numerical solver. Based on the vehicle model, the solver calculates the optimal control values in real time in order to minimize a cost function, which serves as the optimization objective. In comparison to classic control theory approaches, the NMPC is able to predict and control the future behaviour of vehicle states inside of the prediction horizon. Hence, model predictive controllers are very popular for autonomous vehicles. The vehicle model is described as a nonlinear state space, that outlines the vehicle dynamics. We use a kinematic bicycle model, that neglects tire forces, similar to [12] and to the one used inside the SLAM algorithm. The model is implemented in Python as a system of time-continuous differential equations with the vehicle acceleration a_x and the tire angle rate $\dot{\delta}$ as model input. To discretize the model, a 4th order Runge-Kutta integrator is used. In every time step, the NMPC calculates the optimal input vector to solve the optimization objective. Based on the sign of the input acceleration a_x, this value is transformed to either a pneumatic brake pressure or a motor torque value. These control values are published to ROS and then transmitted via CAN to the low-level control devices. Furthermore, these control values are filtered with an IIR-Filter to counter noises and outliers from the whole pipeline. The optimization objective is mathematically described as a quadratic cost function, where the squared difference between the predicted vehicle positions and the reference positions are summed up over the prediction horizon. The reference positions for every time step along the prediction horizon are derived by preprocessing the trajectory similar to [12]. Path points and the velocity profile are used to calculate a time profile, which then is used to extract the exactly-timed reference positions inside the prediction horizon. To solve the optimization objective in real-time, the FORCESPRO NLP solver by Embotech is used [13]. This solver predicts input values, which minimize the cost function. With solving times below 5 milliseconds this solver is very reliable for application inside the ACU. Due to the prediction horizon of $N = 20$ and a time step of 50 ms, the NMPC is able to predict and control the vehicle dynamics one second ahead of the current state using the kinematic bicycle model.

5 Results

Since the benefits and drawbacks of the single modules were already explained in Section 4, this section focuses more on the results and computation times of the whole pipeline.

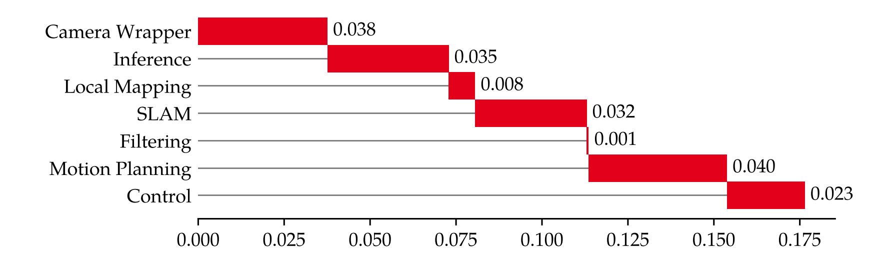


Figure 2: Median processing time of each module of the pipeline in seconds.

Figure 2 shows the median processing time of each module and subsequently of the whole signal processing timeline. It takes about 175ms from the recording of an image until it is represented in the control output. The camera wrapper includes the recording and processing of the image and the encoding into a ROS message. Computation heavy modules like the SLAM and motion planning module have a major share, which can be lowered by using a different parameter set, migrating to a more efficient language like C++ and parallelizing specific computations. The processing time of the control module is misleading, since it is executed with a fixed rate of 20 Hz and thus the median processing time includes idle time. Modules like the inference, local mapping and filtering provide little room for improvement since most of their calculations are carried out with efficient libraries like YOLOv5 or *NumPy*.

Under the assumption that the vehicle velocity is $15\frac{m}{s}$, the total processing time will lead to a loss of 2.65m effective perception range. Since the position dependent modules (filtering, motion planning and control) use separate and newer vehicle position, the control output error due to a wrongfully assumed vehicle position is limited and can be compensated by choosing a corresponding time step of the NMPC.

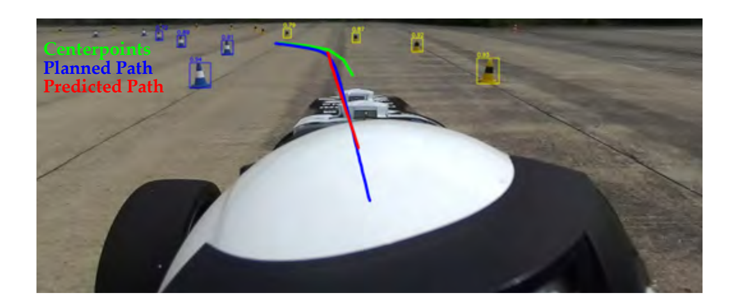


Figure 3: Output of the pipeline visualized on a camera image.

Figure 3 shows the system output visualized on a camera image captured on a testing day with a driver. The detected bounding boxes of the inference module as well as the calculated center points (green), the planned path (blue) and predicted path (purple) are shown.

6 Conclusion and outlook

In this work, the signal processing pipeline for an autonomous race car in the context of Formula Student competitions was presented. Each module was explained in detail, also focusing on its positive and negative aspects regarding computational cost and reliability. The resulting output of the system was visualized and the computational times of each module were analyzed and put into context.

To improve the system in the future, the plan is to improve the calibration method used for the camera perception to enhance the accuracy of the local maps from images. Furthermore, an investment in a lidar with more than 16 channels, Gaussian channel distribution is planned and work is done to correctly integrate it into the system. Finally, the computational times of the motion planning and SLAM modules will be reduced by migrating them to C++.

References

1. Formula Student Germany. Competition Handbook 2022. [Online]. Available: https://www.formulastudent.de/fileadmin/user_upload/all/

2022/rules/FSG22_Competition_Handbook_v1.2.pdf

2. OpenCV. solvePnP. [Online]. Available: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga549c2075fac14829ff4a58bc931c033d
3. Ultralytics, "YOLOv5," Jul. 2021. [Online]. Available: https://github.com/ultralytics/yolov5
4. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767
5. N. Vödisch, D. Dodel, and M. Schötz, "FSOCO: The Formula Student Objects in Context Dataset," *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 12-05-01-0003, 2022.
6. W. Xinyu, "TensorRTx," Jul. 2021, original-date: 2019-11-25T09:01:36Z. [Online]. Available: https://github.com/wang-xinyu/tensorrtx
7. M. Himmelsbach, F. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," 07 2010, pp. 560 – 565.
8. R. Labbe. Filterpy. [Online]. Available: https://filterpy.readthedocs.io/en/latest/
9. T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3149–3154.
10. A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1497–1527, 2020. [Online]. Available: https://doi.org/10.1080/00423114.2019.1631455
11. T. Faulwasser and R. Findeisen, *Nonlinear Model Predictive Path-Following Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 335–343. [Online]. Available: https://doi.org/10.1007/978-3-642-01094-1_28
12. A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, pp. 628–647, 09 2015.
13. A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point... methods for multistage nonlinear nonconvex programs," *International Journal of Control*, pp. 1–17, 2017.

Indoor floorplan estimation from 3D point clouds for *Scan-to-BIM*

Oscar H. Ramírez-Agudelo[1], Antje Alex[2], Lena Schreiber[1], Norman Niemann[1], Edoardo Milana[1], and Christof Hammer[1]

[1] German Aerospace Center (DLR), Institute for the Protection of Terrestrial Infrastructures, Rathausallee 12, 53757 Sankt Augustin, Germany
[2] German Aerospace Center (DLR), Institute for the Protection of Maritime Infrastructures, Fischkai 1, 27572 Bremerhaven, Germany

Abstract Societies depend on the unrestricted availability of their infrastructures. Events such as (natural) disasters, emergencies, or even attacks, could threaten their safety and security. Indoors models provide relevant information that could help in this regard. Their floorplans contain key information such as their location, design, and layout. The architecture, engineering, and construction (*AEC*) community work together to create the respective indoor models within the Building Information Modelling (*BIM*) framework. *BIM* modelling has recently gotten the attention in the computer vision domain. The 1st international *Scan-to-BIM* challenge, organised within the *CVPR 2021* conference, helped to establish research interest and common goals between the *AEC* and computer vision community. In this paper, we introduce a method to estimate floorplans from 3D point cloud data by using the *Scan-to-BIM* dataset. Our work has been developed by using image processing techniques. It does not aim to replace state-of-the-art approaches, which are more elaborate and robust. Instead, it constitutes a non CPU intensive alternative that fairly estimates floorplans for the *Scan-to-BIM* dataset.

Keywords Floorplan estimation, 3D point clouds, *Scan-to-BIM*, data and image processing

1 Introduction

Modern societies depend on the unrestricted availability of their critical infrastructures [1], where buildings constitute main terrestrial infrastructures. They impact our quality of life in many of the same ways as other infrastructures. To protect them from dangers is essential for prosperity and social stability. Events such as (natural) disasters, emergencies, or even attacks, could threaten their safety and security [2]. Therefore, it is important to gather detailed information as well as to provide indoor models [3]. The Institutes for the Protection of Terrestrial and Maritime Infrastructures, subscribed to the German Aerospace Center (DLR), are dedicated to develop concepts and technologies to help to improve the safety and security of critical maritime and terrestrial infrastructures.

The floorplan of buildings becomes a relevant representation of their interiors. In the architecture, engineering, and construction (*AEC*) community, it is standard that such models are done manually, being prone to human errors. Additionally, due to renovation and maintenance, floorplans are often outdated. Moreover, there are other cases, where the floorplans do not even exist [4]. In recent years, with 3D point cloud scanning and technologies such as building information model (*BIM*), the modelling has become a common practice [3]. Although it still encounters computational challenges such as data diversity, accurate geometry, large-scale input, etc. [5], it is currently an active area of research.

Computer vision has already made progress in the detection of walls from buildings [6]. Deep learning has shown promising potential in object detection [7] or in room layout reconstruction tasks such as segmentation and parsing geometry [8–12]. Deep neuronal networks have also been applied to floorplan reconstruction [13–15] (see Section 2).

In this paper, we propose an automatic and light alternative to estimate the 2D floorplan from 3D point cloud data by implementing an image processing approach. This work is structured as follows. Section 2 reviews relevant literature. Section 3 describes the methodology adopted in this work. The results are presented in Section 4 and dis-

cussed in Section 5. The conclusion is presented in Section 6.

2 Relevant works

Computer vision and deep learning tasks have made an effort to reconstruct indoor floorplan environments. In computer vision, some representative works are for instance [14] and [16]. The authors generate the 2D floorplan by using line detection algorithms such as *CANNY* [17] and *RANSAC* [18]. In the latter case, the output model is provided within the *BIM* format. It is important to note, however, that first the reconstruction is only based on walls, i.e. excluding information such as doors and stairs. Second, the reconstruction follows the Manhattan-layout assumption, i.e. the orientation of the floorplan can only be horizontal or vertical.

In deep learning, *Floor-Net* [13] and *FloorPP-Net* [15] are representative frameworks to reconstruct floorplans from 3D point clouds. By using the *Scan-to-BIM* dataset [19], *FloorPP-Net* converts it into point pillars. Then the network learns to predict the corners and the edges, generating the desired floorplan output model. Again the final model is only based on walls. Computer vision and deep learning approaches are currently working to include the information of doors and stairs in their future models. However, due to class imbalance (i.e. $data(wall) \gg data(door)$ or $data(stair)$), this aim is a relatively difficult to accomplish. Besides, due to data pre-processing and algorithm implementation (line detection for computer vision or neuronal networks for deep learning), these frameworks could take up to several minutes to compute ($\sim$5 minutes) and require special graphical processing unit (GPU); making them computing time intensive.

3 Methodology

3.1 Dataset

In this paper, we introduce a method to estimate floorplans from 3D point cloud data by using the *Scan-to-BIM* dataset. The dataset has been obtained from the 1st International *Scan-to-BIM* Challenge [19]. It

was published at the workshop *Computer Vision in the Built Environment* (*CVBE*) as part of the *Computer Vision Pattern Recognition* (*CVPR*) conference in 2021 [20][3]. The dataset includes a wide variety of constructions such as libraries, office labs, short-, medium- and large-offices as well as parking sites. The sample contains a total of 31 buildings with multiple floors each and dozens of rooms on each floor. For 20 buildings it also contains floorplan ground truths. The labels range from wall, door, stair, etc.

3.2 Framework

The methodology developed in this work is based on image processing techniques. The framework is implemented in two-stages.

Algorithm 1

The first-stage consists of the construction of a 2D histogram where all data-points are projected to the $x - y$ plane with the bin size as parameter (*bins*). The histogram returns the x and y edges of the grid (i.e. x_{edges} and y_{edges}) as well as the number of data points per the bi-dimensional bin (H) computed in log-scale.

Algorithm 2

The second-stage computes the floorplan estimation in the following way:

1. The consideration of the output of x_{edges}, y_{edges}, and H computed in the first-stage.

2. The values are normalised with respect to the bin-size to make our method independent of the dimension of the input point cloud.

[3] The CVPR 2022 hosted the 2nd version of the *CVBE*-workshop, where the same dataset has been made available.

3. The ground truth, provided for 20 buildings by the *Scan-to-BIM* dataset, supply the annotation of the labels by segments. A segment is defined by two coordinates; i.e. (x_1, y_1) and (x_2, y_2). For each segment, the class-category (e.g., wall) is assigned to *bins* which distance to that segment is smaller than a *Criterion* (*Crit*) and if the content of that bin, namely H, is larger than the *threshold* (Thr).

The proposed methodology has been applied to several cases. The numeric values of the parameters are: $bins = 1000$, $Crit = 25$, and $Thr \geq 0$. The parameters have been selected after a grid search. They optimise our results without incurring in over-fitting.

3.3 Metrics

We aim to evaluate the position and length of the detected features (e.g. wall) by using the *precision* and *recall*. Based on true-positive (TP), false-positive (FP), and false-negative (FN), we calculate the recall as follows:

$$Precision = \frac{TP}{FP + TP} \tag{1}$$

$$Recall = \frac{TP}{FN + TP} \tag{2}$$

where:

- TP refers to the area of the detected feature (e.g. wall) that is that feature (e.g. wall) in the ground truth.
- FP refers to the area of the detected feature that is not a feature in the ground truth.
- FN is the area that is the feature in the ground truth but is not detected as a wall by the proposed algorithm.

Finally, the Structural Similarity Index ($SSIM$) has also been calculated following the equation 13 of [21]. This is an image quality assessment to compare two images for structural information ranging from 0 (no similarity) to 1 (similar). More details can be found in [21].

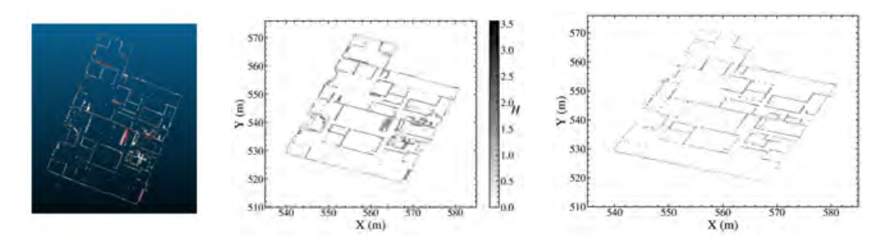


Figure 1: *Small building* of the *Scan-to-BIM* Challenge (see Section 3.2). *Left-panel:* Point Cloud 12_SmallBuilding_02_F1. *Middle-panel:* 2D histogram. Outcome from Sect. Algorithm 1. *Right-panel:* Floorplan estimation. Outcome from Sect. Algorithm 2 (see Sect. 3.2). Labels: walls (black), doors (purple), and stairs (gold).

4 Results

The proposed methodology presented in Sect. 3 has been applied to two different cases. They belong to the training set of *Scan-to-BIM* dataset. Both have ground truth annotations with three categories: wall, door and stair.

4.1 Small building

Figure 1 presents the first experiment. *Left-panel* shows the point cloud of the first floor of a small building with about 17 million data points. First of all, note that this point cloud does not follow the Manhattan layout, i.e. the orientation of the walls of the building does not follow a horizontal or vertical orientation [6]. Second, the data points do not have information of the ceiling or floor. Third, the content of clutter or noise is minimal. Therefore, this becomes an excellent study case.

Middle-panel is the outcome of applying the steps described in Algorithm 1 to the left-panel. It shows the 2D histogram where the maximum value of H is about $H_{max} = 3.5$. *Right-panel* has been constructed by applying the steps described in Algorithm 2 (see Sect. 3.2). There, the floorplan estimation of the *Small building* has been obtained with label annotations. Doors and stairs are rather difficult to retrieve.

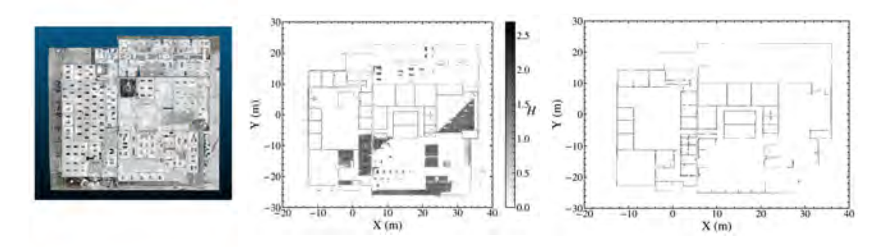


Figure 2: *Office Lab* of the *Scan-to-BIM* Challenge (see Section 3.2). *Left-panel:* Point Cloud 12_SmallBuilding_02_F1. *Middle-panel:* 2D histogram. Outcome from Sect. Algorithm 1. *Right-panel:* Floorplan estimation. Outcome from Sect. Algorithm 2 (see Sect. 3.2) Labels: walls (black), doors (purple), and stairs (gold).

Besides, considering the definition of *precision* and *recall* (see Sect. 3.3), for the feature wall then: *precision* = 1 and *recall* = 0.54 ($TP = 11527$, $FN = 9723$ and $FP = 0$).

4.2 Office Lab

Figure 2 presents the second case. Panel *a)* shows the Office Lab with about 120 million points. This case follows the Manhattan layout. However, it has information of the ceiling. This information needs to be removed. Therefore, this experiment constitutes a much more complex case to study.

Following the work of [16][4], we first proceed with an analysis of height to take out the ceiling as well as the clutter (see sections 3.1 and 3.2 f the mentioned paper). The point cloud is reduced to about six million points. Afterwards, we continue with the implementation of our framework. Panel *b)* shows the 2D histogram. The maximum value of H is about $H_{max} = 2.5$. Panel *c)* shows the floorplan estimation accounting for the label-categories. Once again doors and stairs are rather difficult to retrieve.

As for the metrics defined in Sect. 3.3, for the feature wall then: *precision* = 1 and *recall* = 0.43 ($TP = 16158$, $FN = 21853$ and $FP = 0$).

[4] Repository available in [22].

5 Discussion

5.1 Floorplan: Estimation vs. reconstruction

Figures 3 and 4 compare the floorplan estimation for the *Small building* and *Office Lab*, presented in Figs. 1 and 2, with the ground truth. Table 1 provides statistical insight to our findings. For the *Small building*, the *ratio* between the number of points of the estimated feature divided by the total number of points of the ground truth of that feature, i.e. wall, door and stair are: 54%, 40%, and 6%, respectively (see values in Table 1). Note that the ground truth presents a room around the coordinate $(X, Y) = (578, 556)$ (see Fig. 3 right-panel) that is not present at all in the original point cloud data (see left-panel of Fig. 3). This inconsistency is intrinsic to the dataset. Although it contributes to the discrepancy in our results, it does not explain the difference altogether.

The *ratio* for the *Office Lab* are: 43%, 26% and 4%, respectively. Due to a class imbalance[5], the identification of doors and stairs is limited. This is a well known issue in the literature, where it is common to provide floorplans purely based on walls, e.g. [15,16] (see also Sect. 5.2).

In our approach, the *FPs* are zero for both cases (see Sect. 3). Thus, the *ratio* and *recall* have the same values. The *SSIM* for the *Small building* is 0.91 and for the *Office Lab* is 0.86 (see Table 1), indicating that the floorplan estimation and ground truth, at least for walls, are similar.

5.2 Comparison to other methods

State-of-the-art (*SOTA*) approaches (i.e. computer vision or deep learning) make use of metrics such as *Intersection over Union* (*IoU*), *recall* and *precision*. For instance, the computer vision work of [16] presents great results in their experiments with a *precision* and *recall* over 90%. Similarly, the deep learning work *FloorPP-Net* [15] by using the *Scan-to-BIM* dataset reported a *precision* of 7%, *recall* of 39% and an *IoU* of 12% in the floorplan only based on walls (i.e. without

[5] $data(wall) \gg data(door)$ or $data(stair)$.

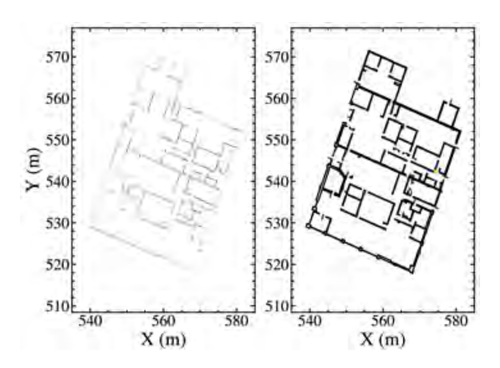


Figure 3: Floorplan estimation (left-panel) vs. the ground truth (right-panel) for the *Small building* presented in Sect. 4.1 where the walls (black), doors (purple) and stairs (gold) are shown.

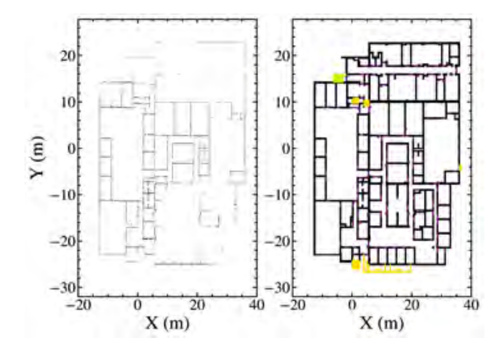


Figure 4: Floorplan estimation (left-panel) vs. the ground truth (right-panel) of the Office Lab presented in Sect. 4.2 where the walls (black), doors (purple) and stairs (gold) are shown.

including information of any other feature such as door or stair).

Computer vision and deep learning are still improving not only in the automatic detection of walls but also in the detection of doors and stairs. However, it is important to note, the calculation could take up to several minutes to compute and often require a special graphical processing unit (GPU).

Table 1: Columns 1-4: Number (#) of data points per class category for the floorplan estimation (our method) vs. Ground truth. The ratio $\left(\frac{\#points\ Estimation}{\#points\ Ground\ Truth}\right)$. Column 5: Result of the Structural Similarity Index (*SSIM*) between the estimation and ground truth.

–	**Wall** (# points)	**Door** (# points)	**Stair** (# points)	***SSIM*** [0,1]
Small building				
Estimation	11527	724	4	0.91
Ground Truth	21250	1791	71	
ratio	54 %	40 %	6 %	–
Office Lab				
Estimation	16158	818	100	0.86
Ground Truth	38011	3181	2698	
ratio	43 %	26 %	4 %	–

Comparing this work to *SOTA*, and by considering that the metric *SSIM* can be understood as a proxy of *IoU*, the results of this work compare well (see also *recall*). Besides, it can be seen as an alternative method to estimate floorplan of buildings. It constitutes a light implementation (i.e. CPU-based), which provides fast and fair floorplan estimation for the *Scan-to-BIM* dataset. By virtue of its simplicity, in the future, its implementation will be extended to other datasets .

6 Conclusion

Based on image processing techniques, we develop an alternative method to estimate floorplan of buildings in the *Scan-to-BIM* dataset. Our method does not aim to replace state-of-the-art approaches, which are more elaborate and robust. It, however, provides a fair automatic floorplan estimation, which may lead to the reconstruction of floorplans.

References

1. "Directive of the european parliament and of the council on the resilience of critical entities," https://eur-lex.europa.eu/resource.html?uri=cellar:

74d1acf7-3f94-11eb-b27b-01aa75ed71a1.0001.02/DOC_1&format=PDF, accessed: 2022-10-03.

2. "Dlr-pi," https://www.dlr.de/pi/en/desktopdefault.aspx/, accessed: 2022-10-03.
3. A. A. Aibinu and E. Papadonikolaki, "Conceptualizing and operationalizing team task interdependences: Bim implementation assessment using effort distribution analytics," *Construction Management and Economics*, vol. 38, no. 5, pp. 420–446, 2020. [Online]. Available: https://doi.org/10.1080/01446193.2019.1623409
4. C. Wang, K. Yong, and C. Kim, "Automatic bim component extraction from point clouds of existing buildings for sustainability applications," vol. 56. Autom. Constr., 2015, pp. 1–13.
5. R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (bim) for existing buildings – literature review and future needs," vol. 38. Autom. Constr., 2014, pp. 109–127.
6. I. Hanagnostopoulos, V. Patraucean, I. Brilakis, and P. Vela, "Detection of walls, floors and ceilings in point cloud data," *Construction Research Congress 2016*, pp. 2302–2311, 2016.
7. Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: exceeding YOLO series in 2021," *CoRR*, vol. abs/2107.08430, 2021. [Online]. Available: https://arxiv.org/abs/2107.08430
8. C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "Planercnn: 3d plane detection and reconstruction from a single image," *CoRR*, vol. abs/1812.04072, 2018. [Online]. Available: http://arxiv.org/abs/1812.04072
9. C. Zou, A. Colburn, Q. Shan, and D. Hoiem, "Layoutnet: Reconstructing the 3d room layout from a single RGB image," *CoRR*, vol. abs/1803.08999, 2018. [Online]. Available: http://arxiv.org/abs/1803.08999
10. C. Sun, C. Hsiao, M. Sun, and H. Chen, "Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation," *CoRR*, vol. abs/1901.03861, 2019. [Online]. Available: http://arxiv.org/abs/1901.03861
11. C. Zou, J. Su, C. Peng, A. Colburn, Q. Shan, P. Wonka, H. Chu, and D. Hoiem, "3d manhattan room layout reconstruction from a single 360 image," *CoRR*, vol. abs/1910.04099, 2019. [Online]. Available: http://arxiv.org/abs/1910.04099
12. C. Yang, J. Zheng, X. Dai, R. Tang, Y. Ma, and X. Yuan, "Learning to reconstruct 3d non-cuboid room layout from a single

RGB image," *CoRR*, vol. abs/2104.07986, 2021. [Online]. Available: https://arxiv.org/abs/2104.07986

13. C. Liu, J. Wu, and Y. Furukawa, "Floornet: A unified framework for floorplan reconstruction from 3d scans," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 203–219.
14. J. Han, M. Rong, H. Jiang, H. Liu, and S. Shen, "Vectorized indoor surface reconstruction from 3d point cloud with multistep 2d optimization," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 177, pp. 57–74, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271621001222
15. Y. Wu and F. Xue, "Floorpp-net: Reconstructing floor plans using point pillars for scan-to-bim," 2021.
16. U. Gankhuyag and J.-H. Han, "Automatic 2d floorplan cad generation from 3d point clouds," *Applied Sciences*, vol. 10, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/8/2817
17. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
18. H. Cantzler, "Random sample consensus (ransac)," *Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh*, 1981.
19. "1st international scan-to-bim challenge," https://cv4aec.github.io/, accessed: 2022-02-04.
20. "Cvpr virtual 2021," https://cvpr2021.thecvf.com, accessed: 2022-10-07.
21. Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
22. "Repository of: Automatic 2d floorplan cad generation from 3d point clouds," https://github.com/joyjo/to-generate-2D-floorplan-CAD-from-3D-point-clouds, accessed: 2022-02-04.

Perspektiveninvariante Inferenz von Eckpunkten in Packmustern von Kartonagen

Perspective invariant inference of corner points in packing patterns of cardboard boxes

Felix Endres, Lucas Reinhart, Tobias Kaupp und Volker Willert

University of Applied Sciences Würzburg-Schweinfurt
Institute Digital Engineering (IDEE)
Ignaz-Schön-Straße 11
97421 Schweinfurt

Zusammenfassung Diese Arbeit befasst sich mit der Inferenz von Eckpunkten von Kartonagen, die in einem regelmäßigen dichten Packmuster flächig angeordnet sind. Als Sensordaten werden ausschließlich 2D Kamerabilder und keine 3D Information benutzt. Die Kartonagen werden aus extremen Perspektiven betrachtet, wie sie typischerweise beim „Blick ins Regal“ für automatisierte Kommissionieraufgaben vorkommen. Ausgehend von vier Eckpunkten einer beliebigen Kartonage wird ein auf dem Doppelverhältnis basierendes Verfahren vorgestellt, das die Eckpunkte aller möglicher benachbarter Kartonagenanordnungen berechnen kann. Des Weiteren wird die Fehlerfortpflanzung unter der Annahme von Eckpunktmessungen mit normalverteiltem Rauschen betrachtet und aus der Fehlerverteilung ein parametrisches Modell für die ortsvarianten 2D Wahrscheinlichkeitsverteilungen aller abgeleiteter Eckpunkte ermittelt.

Schlüsselwörter Mustererkennung, Robotik, perspektivische Invarianten

Abstract This work deals with the inference of corner points of cardboard boxes, which are arranged two-dimensionally in a regular dense packing pattern. Only 2D camera images and no 3D information are used as sensor data. The cardboard boxes are viewed from extreme perspectives, typically encountered when ”looking at the shelf” for automated picking tasks. Starting from

four corners of an arbitrary cardboard box, a method based on the crossratio is presented that can compute the corners of all possible neighboring box arrangements. Furthermore, the error propagation assuming corner point measurements with normally distributed noise is considered and a parametric model for the 2D probability distributions that vary across image location of all derived corner points is obtained from the error distribution

Keywords Pattern recognition, robotics, perspective invariants

1 Einleitung

Das dieser Arbeit zugrundeliegende Forschungsprojekt beschäftigt sich mit der Objekterkennung für Intralogistikanwendungen. Die hier behandelte Problemstellung ergibt sich aus einem Projekt mit einem Industriepartner zur Entwicklung eines mobilen *pick-and-place* Roboters zur automatisierten Kommissionierung diverser Warentypen. Der Roboter soll im Mischbetrieb mit menschlichen Arbeitskräften zur Kommissionierung von Mischpaletten eingesetzt werden, wodurch eine Instrumentierung der Umgebung nur eingeschränkt möglich ist. Daraus ergeben sich insbesondere für die Erkennung der Waren einige Herausforderungen. Im Kommissionierbereich sind sich stark ändernde Lichtverhältnisse durch Sonneneinstrahlung und Verschattung vorherrschend. Zusätzlich können sich die visuellen Objekteigenschaften durch Verschmutzung der Waren verändern. Insbesondere ist aber durch die Palettenhöhe und den daraus resultierenden Blickwinkel auf die Palette teilweise nur eine extreme Perspektive zur Objekterkennung vorhanden (siehe auch Abbildung 1). Da dies ein häufig auftretendes Problem bei der Objekterkennung ist, wurden in der Literatur bereits verschiedene Größen untersucht, die invariant bezüglich perspektivischer Verzeichnung sind [1]. Eine der untersuchten perspektivischen Invarianten ist das Doppelverhältnis, das sich als robustes und genaues Maß erwiesen hat [2]. Des Weiteren wurde das Prinzip des Doppelverhältnisses erweitert, um Flächeninvarianten unter projektiven Abbildungen zu erhalten [3].

Um die Produkte greifen zu können, muss ein Greifpunkt ermittelt werden. Dafür würde eine Rekonstruktion des Packmusters im Bild der



Abbildung 1: links: Originalaufnahme der Kamera, rechts: Auflösungsverlust durch Transformation in eine Draufsicht.

Palette gute Kandidaten für die Greifpunkte liefern, z. B. die Mitte der segmentierten Kartonage. Um eine Rekonstruktion des Packmusters zu erreichen, möchte man Informationen nutzen, die typischerweise verfügbar sind, z. B. Größe und Geometrie der Objekte (Palette, Kartonage, usw.). Eine Möglichkeit zur Rekonstruktion besteht darin, eine Draufsicht der Szene zu erstellen. Ein Beispiel dafür ist in Abbildung 1 zu sehen, wo ein deutlicher Auflösungsverlust und große Interpolationsartefakte für die Kartonagen im hinteren Bereich zu erkennen sind. Das kann zu Fehlern bei der Rekonstruktion des Packmusters der Palette führen. Zudem muss für die Transformation in eine Draufsicht die Pose der Kamera in Bezug zur Oberfläche des Packmusters bzw. die Homographie [4] aus den Bilddaten rekonstruiert werden.

Im folgenden Abschnitt 2 wird ein Ansatz vorgestellt, der direkt auf das Bild ohne vorherige Transformation angewendet werden kann. Unter Verwendung des Doppelverhältnisses, sowie der Breite und der Länge der Kartonagen, geben wir eine Formel zur Berechnung möglicher Eckpunkte von Kartonagen im Packmuster an. Abschnitt 3 zeigt die Ergebnisse unserer Methode, angewandt auf ein Beispielbild. Außerdem werden die Auswirkungen von Messungenauigkeiten bei der Extraktion der Eckpunktkoordinaten der Referenzkartonage auf die Genauigkeit der Eckenerkennung der Kartonagen untersucht. Abschnitt 4 fasst die Ergebnisse zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

2 Herleitung der Inferenz von Eckpunkten

Ausgangspunkt ist eine Ansicht von oben auf eine einzelne Kartonage als Referenz. Die Koordinaten der Eckpunkte werden mit $x_a^{00}, x_b^{00}, x_c^{00}, x_d^{00}$ bezeichnet[1], wobei die hochgestellten Zahlen ein lokales Koordinatensystem für jeden Referenzeckpunkt darstellen. Wenn das Seitenverhältnis der Kartonage bekannt ist, lassen sich die Möglichkeiten berechnen, wie weitere Kartonagen angelegt werden können. Für das Seitenverhältnis 2:1 zeigt Abbildung 2(a) die drei Möglichkeiten, wie eine zweite Kartonage auf der rechten Seite angeordnet werden könnte und Abbildung 2(b) für die obere Seite.

Die Punkte x_i^{10}, x_i^{20} und x_i^{01}, x_i^{02} sind die möglichen Eckpunkte angrenzender Kartonagen in x und y Richtung vom Eckpunkt x_i. Sie werden im Folgenden *inferierte Eckpunkte erster Ordnung* genannt, da sie direkt mit dem Doppelverhältnis berechnet werden können. Die Punkte x_i^{11}, x_i^{12} und x_i^{21}, x_i^{22} werden *inferierte Eckpunkte zweiter Ordnung* genannt, da sie sich wiederum von den Punkten erster Ordnung ableiten lassen.

Daraus ergibt sich die in Abbildung 3 gezeigte Konfiguration von Punkten. Aus dieser Sicht lassen sich die nächsten Eckpunkte direkt berechnen, wenn man die Länge L, die Breite B und die Koordinaten der Eckpunkte x_i^{00} der Referenzkartonage kennt. Bei einer perspektivischen Ansicht sind jedoch die Distanzen zwischen den Eckpunkten von Kartonage zu Kartonage unterschiedlich und verändern sich zusätzlich bei der Veränderung der Perspektive. In diesem Fall kann zur Berechnung der Eckpunkte das Doppelverhältnis genutzt werden.

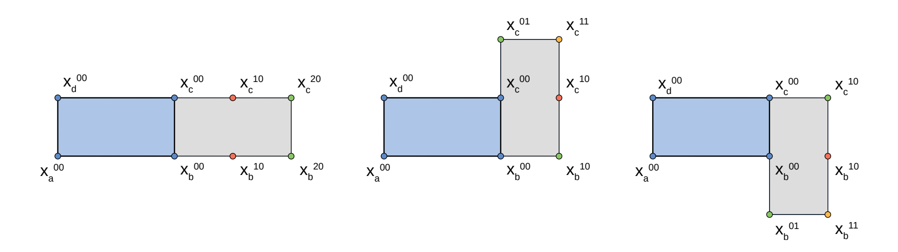
Doppelverhältnisse in Packmustern

Das Doppelverhältnis $[A, B; C, D]$ von vier auf einer Geraden liegenden Punkten A, B, C, D ist definiert durch

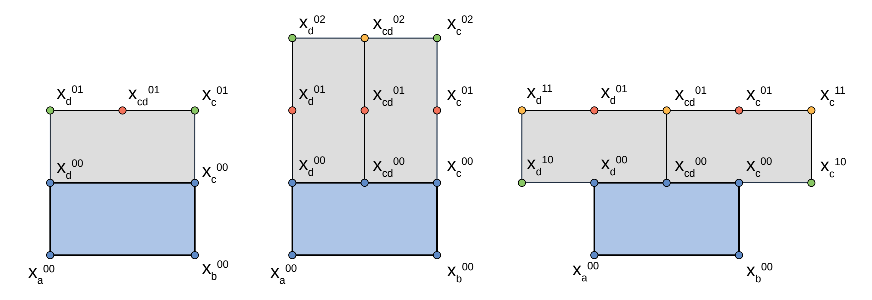
$$[A, B; C, D] := \frac{\overline{AC} \cdot \overline{BD}}{\overline{BC} \cdot \overline{AD}}. \tag{1}$$

Das Doppelverhältnis ist eine projektive Invariante, d.h. es bleibt durch projektive Abbildungen unverändert [5]. Um die nächsten Eckpunkte

[1] Aus Übersichtsgründen werden an manchen Stellen bei den Punkten $x_a^{00}, x_b^{00}, x_c^{00}, x_d^{00}$ die hochgestellten Zahlen weggelassen.



(a) Anordnungen von zwei Kartonagen mit Seitenverhältnis 2:1, die nebeneinander liegen.



(b) Anordnungen von Kartonagen mit Seitenverhältnis 2:1, die übereinander liegen.

Abbildung 2: Grüne Punkte sind die korrekten inferierten Eckpunkte erster Ordnung. Rote Punkte sind Eckpunkte erster Ordnung aus anderen möglichen Konfigurationen. Gelbe Punkte sind inferierte Punkte zweiter Ordnung.

unter einer perspektivischen Ansicht abzuleiten, werden zuerst aus der bekannten Konfiguration (siehe Abbildung 4) mit beliebigen L, B und x_a, x_b, x_c, x_d die vier möglichen Doppelverhältnisse r_1, r_2, r_3, r_4 berechnet:

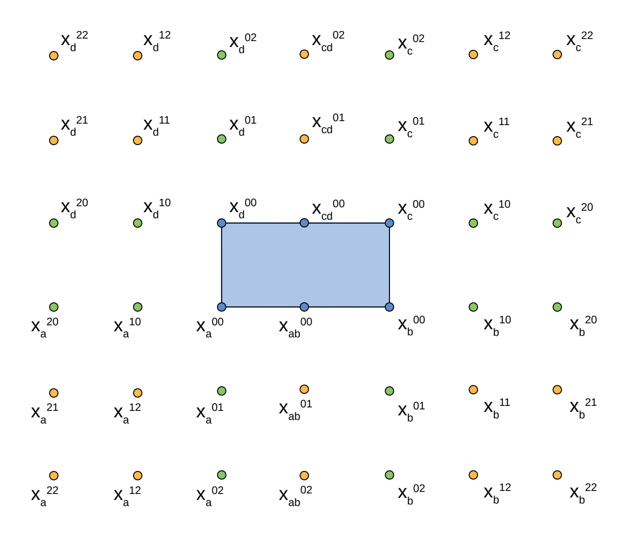


Abbildung 3: Eckpunkte für alle Konfigurationen von Kartonagen mit Seitenverhältnis 2:1. Blaue Punkte sind Teil der Referenzkartonage. Grüne Punkte sind inferierte Punkte erster Ordnung. Gelbe Punkte sind inferierte Punkte zweiter Ordnung.

$$\begin{aligned}
r_1 &= [x_a^{00}, x_{ab}^{00}; x_b^{00}, x_b^{10}] = [x_b^{00}, x_{ab}^{00}; x_a^{00}, x_a^{10}] = [x_d^{00}, x_{cd}^{00}; x_c^{00}, x_c^{10}] \\
&= [x_c^{00}, x_{cd}^{00}; x_d^{00}, x_d^{10}] = \frac{L \cdot (L/2 + B)}{L/2 \cdot (L + B)} = \frac{L + 2B}{L + B}\,, \qquad (2) \\
r_2 &= [x_a^{00}, x_{ab}^{00}; x_b^{00}, x_b^{20}] = [x_b^{00}, x_{ab}^{00}; x_a^{00}, x_a^{20}] = [x_d^{00}, x_{cd}^{00}; x_c^{00}, x_c^{20}] \\
&= [x_c^{00}, x_{cd}^{00}; x_d^{00}, x_d^{20}] = \frac{L \cdot (L/2 + L)}{L/2 \cdot (L + L)} = \frac{3}{2}\,, \qquad (3) \\
r_3 &= [x_a^{00}, x_{da}^{00}; x_d^{00}, x_d^{01}] = [x_d^{00}, x_{da}^{00}; x_a^{00}, x_a^{01}] = [x_b^{00}, x_{bc}^{00}; x_c^{00}, x_c^{01}] \\
&= [x_c^{00}, x_{bc}^{00}; x_b^{00}, x_b^{01}] = \frac{B \cdot (B/2 + B)}{B/2 \cdot (B + B)} = \frac{3}{2}\,, \qquad (4) \\
r_4 &= [x_a^{00}, x_{da}^{00}; x_d^{00}, x_d^{02}] = [x_d^{00}, x_{da}^{00}; x_a^{00}, x_a^{02}] = [x_b^{00}, x_{bc}^{00}; x_c^{00}, x_c^{02}] \\
&= [x_c^{00}, x_{bc}^{00}; x_b^{00}, x_b^{02}] = \frac{B \cdot (B/2 + L)}{B/2 \cdot (B + L)} = \frac{B + 2L}{B + L}\,. \qquad (5)
\end{aligned}$$

Interessanterweise ergeben sich nur drei unterschiedliche Werte, wobei r_1 und r_4 von L und B abhängen und $r_2 = r_3 = 3/2$ identisch und unabhängig vom Seitenverhältnis der Kartonage sind.

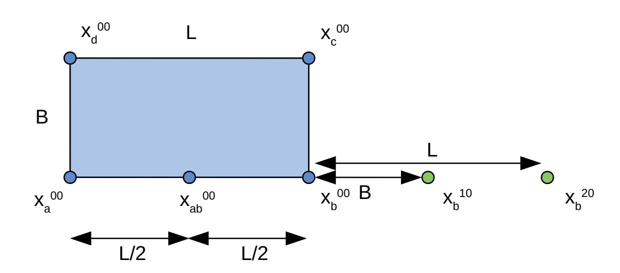


Abbildung 4: Mit bekannter Länge L und Breite B der Referenzkartonage können vier mögliche Doppelverhältnisse berechnet werden (siehe Text).

Eckpunktberechnungen

Als nächstes wird die Konfiguration aus einer beliebigen Perspektive betrachtet, wie in Abbildung 5 dargestellt. Unter Verwendung homogener Koordinaten können Hilfspunkte x_e, x_f und x_g aus den Gleichungen 6 - 8 berechnet werden. Hierbei steht $\times$ für das dreidimensionale Kreuzprodukt und $\mathbf{x_i}$ steht für den Koordinatenvektor des Punktes x_i . Jede Gerade durch die Punkte x_i und x_j wird durch einen Vektor $\mathbf{l_{ij}}$ parametrisiert:

$$\mathbf{l_{ac}} = \mathbf{x_a} \times \mathbf{x_c}, \qquad \mathbf{l_{bd}} = \mathbf{x_b} \times \mathbf{x_d}, \qquad \mathbf{x_e} = \mathbf{l_{ac}} \times \mathbf{l_{bd}}\,, \tag{6}$$

$$\mathbf{l_{ab}} = \mathbf{x_a} \times \mathbf{x_b}, \qquad \mathbf{l_{cd}} = \mathbf{x_c} \times \mathbf{x_d}, \qquad \mathbf{x_g} = \mathbf{l_{ab}} \times \mathbf{l_{cd}}\,, \tag{7}$$

$$\mathbf{l_{ad}} = \mathbf{x_a} \times \mathbf{x_d}, \qquad \mathbf{l_{bc}} = \mathbf{x_b} \times \mathbf{x_c}, \qquad \mathbf{x_f} = \mathbf{l_{ad}} \times \mathbf{l_{bc}}\,. \tag{8}$$

Damit benachbarte Punkte über das Doppelverhältnis nur aus den vier Referenzpunkten inferiert werden können, werden die vier Seitenhalbierende $x_{ab}, x_{cd}, x_{bc}, x_{da}$ als Hilfspunkte eingeführt. Die Seitenhalbierenden x_{ij}, $i \neq j \in \{a, b, c, d\}$ werden konstruiert, indem die Gerade auf dem einer der Fluchtpunkte x_f, x_g und der Mittelpunkt der Refe-

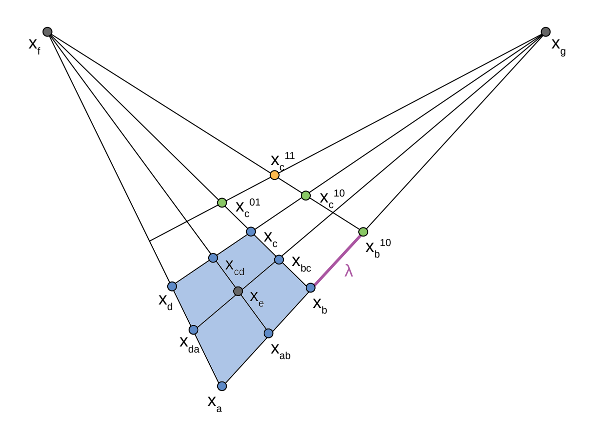


Abbildung 5: Konfiguration bei beliebiger Perspektive.

renzkartonage x_e liegt, mit einer Seite des Vierecks geschnitten wird:

$$\mathbf{x_{ab}} = \mathbf{l_{fe}} \times \mathbf{l_{ab}}, \qquad \mathbf{x_{bc}} = \mathbf{l_{ge}} \times \mathbf{l_{bc}}\,, \tag{9}$$

$$\mathbf{x_{cd}} = \mathbf{l_{fe}} \times \mathbf{l_{cd}}, \qquad \mathbf{x_{da}} = \mathbf{l_{ge}} \times \mathbf{l_{da}}\,. \tag{10}$$

Mit den Seitenhalbierenden und dem Doppelverhältnis können wir dann den Abstand λ zu einem Nachbarpunkt x_b^{10} berechnen:

$$r_1 = \frac{\|\mathbf{x_a} - \mathbf{x_b}\| \|\mathbf{x_{ab}} - \mathbf{x_b^{10}}\|}{\|\mathbf{x_{ab}} - \mathbf{x_b}\| \|\mathbf{x_a} - \mathbf{x_b^{10}}\|} = \frac{\|\mathbf{x_a} - \mathbf{x_b}\| \left(\|\mathbf{x_{ab}} - \mathbf{x_b}\| + \lambda\right)}{\|\mathbf{x_{ab}} - \mathbf{x_b}\| \left(\|\mathbf{x_a} - \mathbf{x_b}\| + \lambda\right)},$$

$$\Rightarrow \lambda = \frac{(1 - r_1)\left(\|\mathbf{x_{ab}} - \mathbf{x_b}\| \|\mathbf{x_a} - \mathbf{x_b}\|\right)}{r_1 \|\mathbf{x_{ab}} - \mathbf{x_b}\| - \|\mathbf{x_a} - \mathbf{x_b}\|}. \tag{11}$$

Schließlich ergeben sich die Koordinaten des Punktes x_b^{10} wie folgt:

$$\mathbf{x_b^{10}} = \mathbf{x_b} + \lambda \frac{\mathbf{x_b} - \mathbf{x_a}}{\|\mathbf{x_b} - \mathbf{x_a}\|}. \tag{12}$$

Auf die gleiche Weise ist es möglich alle anderen Eckpunkte erster Ordnung zu berechnen. Den Eckpunkt zweiter Ordnung x_c^{11} erhält man durch

$$\mathbf{l_{c^{10}f}} = \mathbf{x_c^{10}} \times \mathbf{x_f}, \quad \mathbf{l_{c^{10}f}} = \mathbf{x_c^{01}} \times \mathbf{x_g} \tag{13}$$

$$\Rightarrow \mathbf{x_c^{11}} = \mathbf{l_{c^{10}f}} \times \mathbf{l_{c^{01}g}}. \tag{14}$$

Durch das Schneiden von Geraden, die aus den inferierten Eckpunkten erster Ordnung mit den Fluchtpunkten gebildet werden, können so auch alle anderen Eckpunkte zweiter Ordnung ermittelt werden.

3 Statistische Analyse und Parametrisches Modell

In der Abbildung 6 wurde die im Abschnitt 2 vorgestellte Methode auf ein Beispielbild angewandt. Dabei wurden die Eckpunkte x_a, x_b, x_c, x_d einer Kartonage im Bild als gemessene Referenzpunkte angenommen. Dann wurden auf Basis einer bivariaten Normalverteilung $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ mit Mittelwertvektor $\boldsymbol{\mu} = \mathbf{x_i}$ und Kovarianzmatrix $\boldsymbol{\Sigma}$, die der Einheitsmatrix entspricht, für jeden Eckpunkt 2000 verrauschte Eckpunkte berechnet. Anschließend wurden für alle 2000 Konfigurationen die Eckpunkte erster und zweiter Ordnung berechnet.

Daraufhin wurden mit Hilfe des Expectation–Maximization Algorithmus [6] die resultierenden Verteilungen der inferierten Eckpunkte durch eine bivariate Normalverteilung approximiert (siehe Abbil-



Abbildung 6: Mögliche Konfiguration von Kartonagen. Blau gefärbt sind die 2000 verrauschten Referenzeckpunkte. Grün sind die daraus resultierenden inferierten Eckpunkte erster Ordnung und orange die inferierten Eckpunkte zweiter Ordnung.

dung 8). Es ist deutlich zu erkennen, dass die Orientierungen der Normalverteilungen sehr gut mit der Orientierung der Verbindungslinien zwischen dem Mittelpunkt der Referenzkartonage und dem jeweiligen inferierten Punkt übereinstimmen. Abbildung 7 zeigt die Standardabweichungen σ_1, σ_2 der ersten und zweiten Hauptkomponente aller bivariaten Normalverteilungen in Abhängigkeit von der Distanz des urspünglichen inferierten Punktes zum Mittelpunkt der Kartonage x_e. Für die erste Hauptkomponente kann die Abhängigkeit durch ein Polynom zweiter Ordnung $p(x)$ approximiert werden, für die zweite Komponente ist die Korrelation durch eine Gerade $l(x)$ beschrieben. Hieraus lässt sich ein Modell für den Einfluss des normalverteilten Rauschens auf die inferierten Punkte ableiten. Sei x_i^{nm} ein von der ursprünglichen Konfiguration inferierter Punkt und $\mathbf{u}$ der Vektor

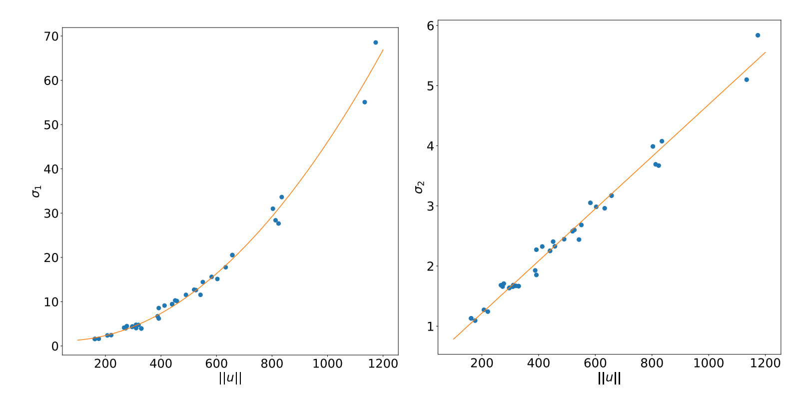
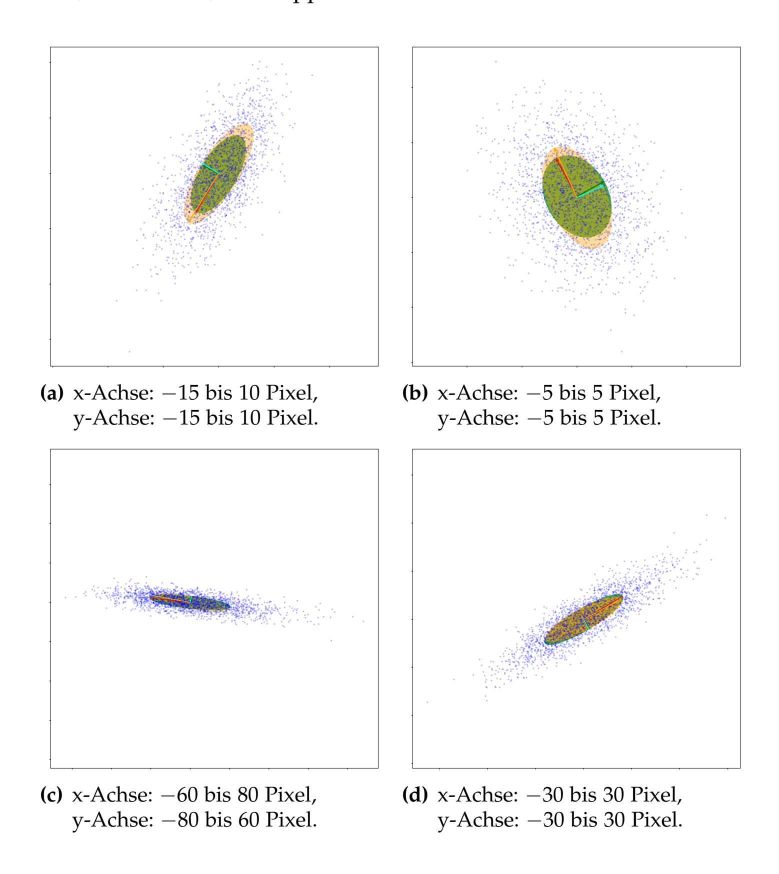


Abbildung 7: Abhängigkeit der Standardabweichungen σ_1, σ_2 der beiden Hauptkomponenten von der Distanz $\|u\|$ in Pixel.

von x_e zu x_i^{nm}, d.h. $\mathbf{u} = \mathbf{x_i^{nm}} - \mathbf{x_e}$. Als nächstes definiert man den Vektor $\mathbf{v} := (-u_2, u_1)$, sodass $\mathbf{u}$ und $\mathbf{v}$ senkrecht aufeinander stehen. Dann ergibt sich ein parametrisches Modell für die Verteilung von x_i^{nm} in Abhängigkeit von x_e, das einer bivariaten Normalverteilung $\mathcal{N}(\boldsymbol{\mu}_i^{nm}, \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{nm})$ mit $\boldsymbol{\mu}_i^{nm} = \mathbf{x_i^{nm}}$ und

$$\boldsymbol{\Sigma}_\mathbf{i}^\mathbf{nm} = \left(\frac{\mathbf{u}}{\|\mathbf{u}\|}, \frac{\mathbf{v}}{\|\mathbf{v}\|}\right) \begin{pmatrix} p(\|\mathbf{u}\|)^2 & 0 \\ 0 & l(\|\mathbf{u}\|)^2 \end{pmatrix} \left(\frac{\mathbf{u}}{\|\mathbf{u}\|}, \frac{\mathbf{v}}{\|\mathbf{v}\|}\right)^\mathsf{T} \tag{15}$$

entspricht. In Abbildung 8 sind beispielhaft für vier inferierte Punkte die Verteilungen mit den 2σ Konturen der Normalverteilung, welche durch den EM-Algorithmus berechnet wurde und der Normalverteilung, die mit dem Modell ermittelt wurde, dargestellt. Sowohl das Modell der Orientierung, als auch das lineare Modell der Standardabweichung entlang der zweiten Hauptkomponente passen sehr gut mit der normalverteilten Statistik der Datenpunkte überein. Bei der Standardabweichung der ersten Hauptkomponente ergeben sich teilweise vertretbare Abweichungen. In zukünftigen Arbeiten sollte das hier vorgestellte datengetriebene Modell durch eine stringente Berechnung der Fehlerfortpflanzung mittels der Formel (12) unter Annahme von normalverteilten Koordinaten der Referenzeckpunkte verifiziert werden.



(a) x-Achse: −15 bis 10 Pixel, y-Achse: −15 bis 10 Pixel.

(b) x-Achse: −5 bis 5 Pixel, y-Achse: −5 bis 5 Pixel.

(c) x-Achse: −60 bis 80 Pixel, y-Achse: −80 bis 60 Pixel.

(d) x-Achse: −30 bis 30 Pixel, y-Achse: −30 bis 30 Pixel.

Abbildung 8: Verteilungen für vier inferierte Eckpunkte. Die grüne Ellipse ist die 2σ Kontur der gefitteten Normalverteilung, die orangene Ellipse ist die 2σ Kontur der aus dem Modell berechneten Normalverteilung. Der rote und grüne Vektor zeigt die Vorzugsrichtungen der EM-Normalverteilung, orange und cyan, die der Modell-Normalverteilung.

4 Zusammenfassung

Es konnte gezeigt werden, dass im Prinzip alle möglichen Eckpunkte rechteckiger Elemente, die zu einem regelmäßigen, dicht besetzten, flächigen Muster zusammengesetzt werden, ausgehend von vier gemessenen Eckpunkten eines beliebigen Elements aus dieser Anord-

nung, über das Doppelverhältnis und das Wissen über die Abmaße des Rechtecks berechnet werden können. Insbesondere bei extremer Perspektive können sich Vorteile ergeben, da trotz starker Abnahme der Bildauflösung der einzelnen projizierten Rechtecke mit zunehmender Distanz zur Kamera, die Lage der Eckpunkte genau berechenbar ist. Ungenauigkeiten bei der Messung der Eckpunktkoordinaten des Referenzrechteckes pflanzen sich weniger stark fort, je größer der Abstand und kleiner die Auflösung in der Projektion sind.

Damit kann dieses Verfahren nicht nur zur Lageerkennung von Kartonagen, sondern auch auf andere Muster, die aus rechteckigen Elementen bestehen adaptiert werden, wie beispielsweise Parkettböden, Mauerwerk oder schachbrettartige Kalibriermuster. Die Fehlerverteilung lässt sich ziemlich genau mit einer ortsvarianten zweidimensionalen Normalverteilung approximieren, wobei die ortsabhängigen Werte des Erwartungswertvektors und der Kovarianzmatrix in Abhängigkeit vom Mittelpunkt des Referenzrechtecks und der Koordinaten des zu inferierenden Eckpunktes berechnet werden können. Damit ergibt sich ein vollständiges probabilistisches Modell, das beispielsweise als Potentialfunktion in einem Markovschen Zufallsfeld zur Bestimmung des vollständigen Packmusters von Kartonagen benutzt werden kann.

Literatur

1. I. Weiss, “Geometric invariants and object recognition,” *International Journal of Computer Vision*, vol. 10, no. 3, pp. 207–231, Jun. 1993.
2. D. Forsyth, J. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, “Invariant descriptors for 3d object recognition and pose,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 971–991, 1991.
3. L. Nielsen and G. Sparr, “Projective area-invariants as an extension of the cross-ratio,” *CVGIP: Image Understanding*, vol. 54, no. 1, pp. 145–159, 1991.
4. Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.
5. J. N. Cederberg, *A Course in Modern Geometries*, 2nd ed., ser. Undergraduate Texts in Mathematics. Springer-Verlag New York, 2001.
6. C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

Orts- und zeitaufgelöste bildbasierte Bestimmung der Brechungsindexverteilung bei der additiven Fertigung optischer Komponenten

Locally and temporally resolved image-based determination of the refractive index distribution during additive manufacturing of optical components

Manuel Rank und Andreas Heinrich

Aalen University, Center for Optical Technologies
Anton Huber Straße 21, 73430 Aalen

Zusammenfassung Der Brechungsindex stellt eine wesentliche Eigenschaft einer optischen Komponente dar. Dabei ist es oft wünschenswert diesen material- und prozessabhängigen Parameter 2dimensional mit einer möglichst hohen Ortsauflösung zu erfassen. Dies trifft auch für den Fall additiv gefertigter Optiken zu. Solche Komponenten können z.B. mit Hilfe der Photopolymerisation realisiert werden. Dabei wird mit UV Strahlung die Aushärtung eines flüssigen Polymer ermöglicht. Wesentlich ist dabei, dass der Aushärtegrad und der damit in Zusammenhang stehende Brechungsindex von den Bestrahlungseigenschaften abhängt. Somit ergibt sich neben dem materialabhängigen Brechungsindex auch eine prozessbedingte Abhängigkeit des Brechungsindexes. Im Umkehrschluss können damit über eine Messung der 2dimensionalen Brechungsindexverteilung auf den Aushärtegrad einer Probe geschlossen und Prozessparameter definiert werden.
In diesem Beitrag soll eine Möglichkeit der orts- und zeitaufgelösten Vermessung des Brechungsindexes während und nach der UV Aushärtung von Polymeren vorgestellt und diskutiert werden. Die Grundlage dafür bildet ein Messansatz basierend auf Totalreflexion. Gewonnen werden mit dem Messaufbau Bilder, welche die Information des lokal vorliegenden Brechungsin-

dex enthalten. Diese können entweder analytisch, oder mit Hilfe eines neuronalen Netzes ausgewertet werden.

Schlüsselwörter Brechungsindex, optische Messung, additive Fertigung, Stereolithografie

Abstract The refractive index represents an essential property of an optical component. It is often desirable to measure this material- and process-dependent parameter 2-dimensionally with the highest possible spatial resolution. This also applies to the case of additively manufactured optics. Such components can be realized, for example, with the aid of photopolymerization. In this process, UV radiation is used to cure a liquid polymer. It is important to note that the degree of curing and the associated refractive index depend on the irradiation properties. Thus, in addition to the material-dependent refractive index, there is also a process-dependent dependence of the refractive index. Conversely, a measurement of the 2-dimensional refractive index distribution can be used to infer the degree of curing of a sample and subsequently to define process parameters.
In this paper, a possibility of spatially and temporally resolved measurement of the refractive index during and after UV curing of polymers will be presented and discussed. The basis for this is a measurement approach based on total internal reflection. Images are obtained with the measurement setup, which contain the information of the locally present refractive index. These can be evaluated either analytically or with the help of a neural network.

Keywords Refractive index, optical metrology, additive manufacturing, stereolithography

1 Einleitung

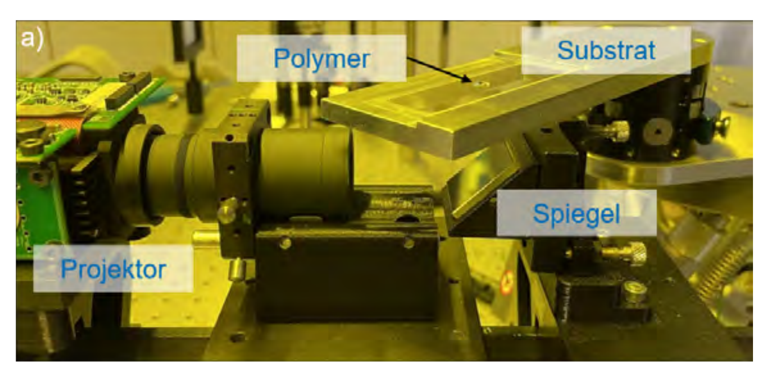
In vielen Bereichen und Anwendungen ist eine ortsaufgeöste Vermessung des Brechungsindexes wünschenswert. Dies trifft auch auf die additive Fertigung von optischen Komponenten zu und gilt vor allem im speziellen Fall der Photopolymerisation von flüssigen Harzen durch UV Bestrahlung.

Ein entsprechender Versuchsaufbau, welcher die Untersuchung der Aushärtung von photonsensitiven Polymeren ermöglicht, ist in Abbildung 1 gezeigt. Wie in Abbildung a) zu erkennen, wird das Licht mit Hilfe eines UV Projektors (Wintech Pro4500, $\lambda = 450nm$, 912x1140 Pixel) in Form einer Pixelmaske erzeugt (Intel DLP System). Ein einzelnes Pixel weist dabei eine Kantenlänge von 35µm auf. Das eingestellte Maskendesign wird über einen Umlenkspiegel und ein Objektträgerglas (Substrat) auf das flüssige Polymer abgebildet, um dieses auszuhärten (s. Abbildung b). Dabei härtet lediglich das Polymer in den belichteten Bereichen aus, wobei es auch zu einem Übersprechen der einzelnen aktiven Pixel in Nachbarbereiche kommt. Da der sich einstellende finale Brechungsindex vom Aushärtegrad des Polymers sensitiv abhängt, ist direkt einsichtig, dass je nach Maske sich lokal unterschiedliche Brechungsinidzes ergeben können. Ein mögliches Maskendesign ist in Abb. c) dargestellt. Hierbei wurde jede zweite Pixelspalte aktiviert. Zusätzlich ist auch zu erkennen, dass fertigungsbedingt die einzelnen Pixel in der Mitte einen „schwarzen Punkt", also einen Bereich aufweisen, in dem sie kein UV Licht auf die Probe senden können. Auch dies führt aufgrund der Inhomogenität in der Bestrahlung zu einer Inhomogenität in der Brechungsindexverteilung.

Da also die lokale UV Bestrahlungsstärke den lokalen Aushärtegrad des Polymers bestimmt, und dieser wiederum mit dem sich dabei ergebenden Brechungsindex korreliert, kann so über die Prozessparameter die lokale Eigenschaft einer additiv gefertigten Optik manipuliert werden. Um für eine gewünschte Brechungsindexverteilung die richtigen Prozessparameter einstellen zu können, ist vorab der nichtlineare Zusammenhang zwischen Prozessparameter (lokale Bestrahlungsstärke) und lokaler Brechungsindex zu erarbeiten. Dies erfordert eine orts- und zeitaufgelöste Messung des Brechungsindexes während der Bestrahlung, welche hier diskutiert werden soll.

2 Experimenteller Aufbau

Der realisierte Messaufbau zur örtlichen und zeitlichen Vermessung des Brechungsindexes ist in Abbildung 2 dargestellt und stellt eine Weiterentwicklung ([1], [2]) des in der Literatur diskutierten scanning focused refractive index microscopes dar [3]. Auf einem Prisma



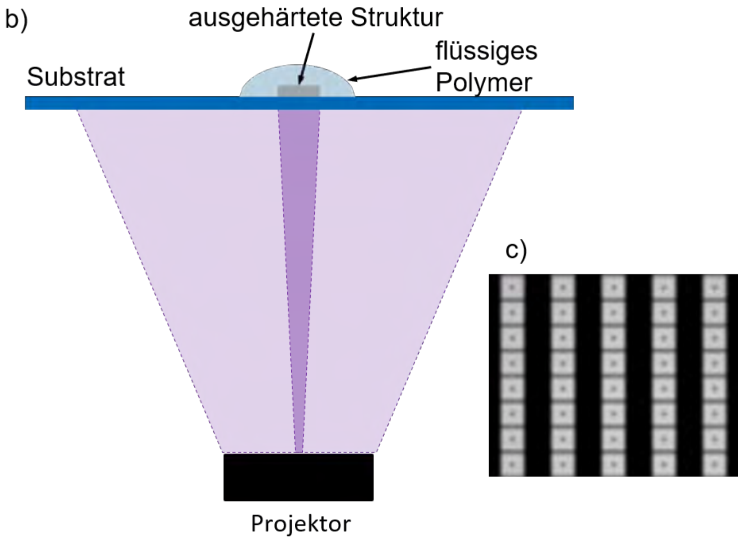


Abbildung 1: a) Experimenteller Aufbau zur Untersuchung der UV Aushärtung von Photopolymeren; b) schematische Darstellung des Lichtwegs; c) Beispiel einer Pixel-basierten Beleuchtung: alternierend wurde eine Pixelspalte an bzw. aus geschaltet. In der Mitte eines jeden Pixels ist ein fertigungsbedingter „Totbereich" zu erkennen.

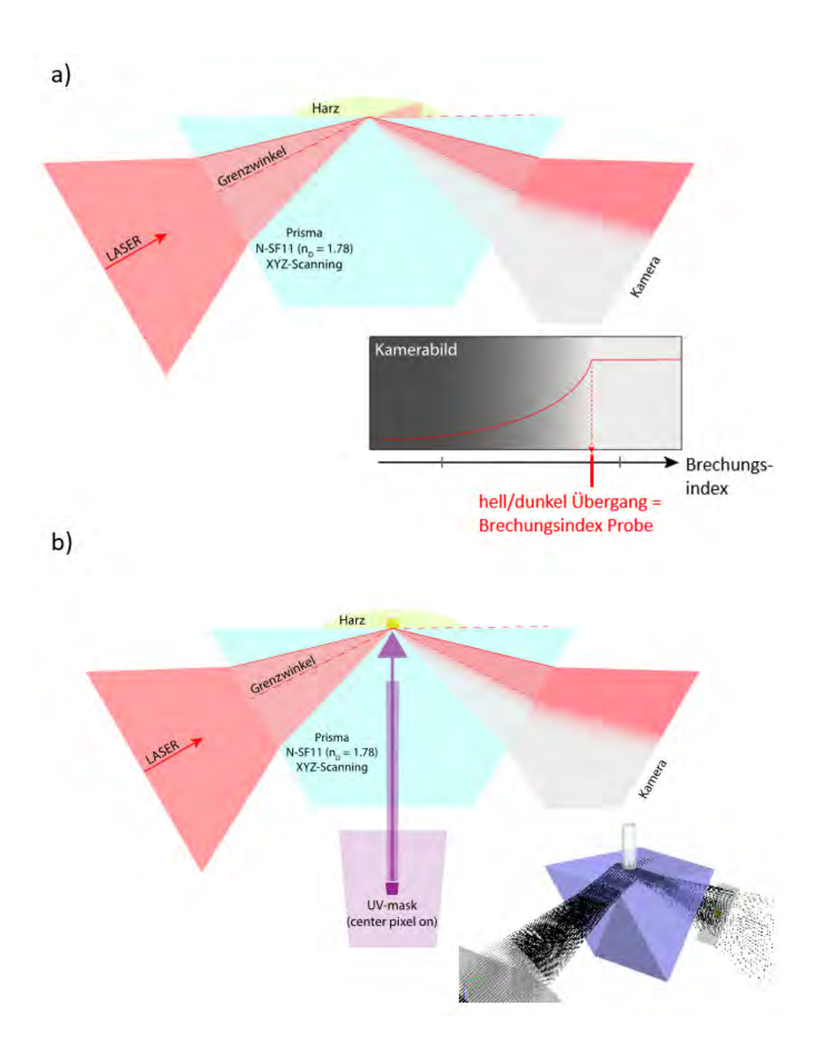


Abbildung 2: a) SFRIM – optisch Vermessung des Brechungsindexes basierend auf Totalreflexion. b) Weiterentwicklung: LineSFRIM – zeitlich und ortsaufgelöste Vermessung des Brechungsindexes während der UV Polymerisation.

(Abbildung a) / SFRIM Verfahren) befindet sich die zu untersuchende Probe – z.B. ein flüssiges Polymer, dessen Brechungsindex bestimmt werden soll oder eine plane Probe, welche über eine „Index-matching“ Flüssigkeit in Kontakt mit dem Prisma gebracht wird. Ein Laserstrahl wird auf die Grenzfläche zwischen Probe und Prisma fokussiert. Aufgrund der Fokussierung treffen Lichtstrahlen aus einem bestimmten Winkelbereich auf die Probe. Diese Lichtstrahlen werden entsprechend der Bedingung für Totalreflexion entweder total reflektiert, oder aus dem Prisma ausgekoppelt. Der kritische Winkel, unter dem Totalreflexion auftritt, ist dabei gegeben durch:

$$\theta_c = arcsin\left(\frac{n_{Probe}}{n_{Prisma}}\right)$$

Durch Bestimmung des kritischen Winkels kann also auf den Brechungsindex der Probe zurück geschlossen werden.
Wird nun über einen Kamera Chip das total reflektierte Licht aufgenommen, so kann aus der Position der Kante des hell – dunkel Übergangs der Totalreflexionswinkel und damit direkt der lokale Brechungsindex (gemittelt über die Fläche des Fokuspunktes des Laserstrahls - hier Durchmesser ca. 2µm) der Probe bestimmt werden. Für eine quantitative Auswertung ist dabei vorab eine Kalibrierung des Systems (bzw. der hell - dunkel Kante) mit Proben mit bekanntem Brechungsindex notwendig. Hierzu wurde eine Bandbreite von kommerziellen Index-Matching Gele verwendet.
Grundsätzlich kann so im Fokuspunkt des Lasers der Brechungsindex der Probe bestimmt werden. Um eine ortsaufgelöste Messung zu ermöglichen, wird über einen x-y Verschiebetisch das Prisma relativ zum Laserstrahl bewegt. Letztendlich erhält man dadurch eine 2dimensionale Vermessung der Brechungsindexverteilung an der Oberfläche der Probe mit einer Auflösung von 2µm. Eine zeitlich aufgelöste Messung der Aushärtung des Polymers ist in dieser Konfiguration nicht möglich.

Für eine zeitaufgelöste Vermessung der Brechungsindexverteilung während der Aushärtung wurde das System modifiziert (s. Abb. b / LineSFRIM Verfahren). In diesem Fall wird statt eines punktförmigen Fokus ein linienförmiger Fokus auf das Interface Prisma / Probe gebracht. Es ergibt sich somit die Möglichkeit einer ortsaufgelösten Vermessung der Brechungsindexverteilung entlang der Fokusline. Des Weiteren wird der oben erwähnte UV Projektor zur Aushärtung

des Polymers mit in den Aufbau integriert, um das Polymer auszuhärten und parallel die Brechungsindexänderung entlang der Fokusline während des Aushärtevorgangs untersuchen zu können. Die zeitliche Auflösung des Systems ist dabei durch die Framerate der Kamera bestimmt.

3 Ergebnisse

In Abbildung 3 sind Ergebnisse der beiden Verfahren SFRIM und LineSFRIM gegenübergestellt. Wie im jeweiligen rechten Bildabschnitt zu erkennen, wurde als Maske für den Projektor ein alternierendes Muster aus jeweils 5 angeschalteten Pixelreihen (helle Reihen) und 5 ausgeschalteten Pixelreihen (dunkle Bereiche) zur Aushärtung verwendet. Jedes aktive Pixel führt dabei zu einer lokalen Aushärtung des UV Polymers am Interface des Prismas und damit zu einer Änderung des Brechungsindexes.

Betrachtet werden soll zunächst die punktförmige Messung basierend auf der SFRIM Methode, welche in Abbildung a) gezeigt ist. Das Ergebnisbild ist durch ein abscannen der Probe in y-Richtung (senkrecht zu den aktivierten Pixelreihen / gelbe Punkte) über einen Bereich von ca. 0,5mm entstanden. Jede horizontale Zeile im Ergebnisbild entspricht dabei einer Messung für einen bestimmten y-Wert auf der Probe. In dieser Messung wird der hell-dunkel Übergang mit Hilfe der Kamera aufgenommen und als Zeile im Ergebnisbild dargestellt. Die x-Achse im Ergebnisbild entspricht damit dem Brechungsindex. Wie in der Abbildung zu erkennen, zeigt sich entlang der punktierten Linie eine Erhöhung des Brechungsindexes (hell - dunkel Kante „wandert nach rechts“) für den Fall, dass eine Aushärtung der Probe stattfindet. In den unbeleuchteten Bereichen bleibt die Position bzw. der Brechungsindex unverändert.

In Abbildung b) wird der Versuch wiederholt, nur wird zur Vermessung der Brechungsindexverteilung das LineSFRIM Verfahren verwendet. Aufgrund dessen, dass der Linienfokus (gelbe Linie) den kompletten Bereich überstreicht, handelt es sich hier um eine einzige Messung, d.h. das dargestellte Ergebnisbild entspricht einer einzigen Aufnahme. Es ist erneut eine identische Ausprägung der Brechungsindexverteilung zu verzeichnen. Allerdings ist dabei auch ein wesentlicher

Nachteil dieser 20x schnelleren „one shot" Methode zu erkennen: die örtliche Auflösung ist reduziert. Dennoch ist sie ausreichend, um zeitliche Vorgänge während der UV Bestrahlung zu untersuchen.

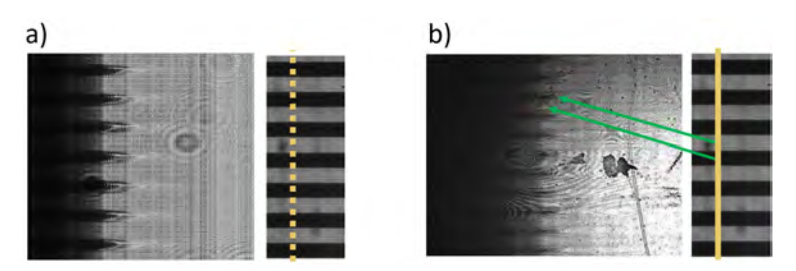


Abbildung 3: a) SFRIM und b) LineSRIM Vermessung der Brechungsindexverteilung nach der Aushärtung eines Polymers mit Hilfe eines Linienmusters einer UV DMD Beleuchtung.

In Abbildung 4 ist eine zeitaufgelöste Messung der Brechungsindexverteilung während der Aushärtung mit Hilfe der LineSFRIM Methode gezeigt. Abbildung a) zeigt die schematische Anordnung in Seiten- und Draufsicht. Auf das Polymer wird in diesem Fall ein UV Puls (violett) mit Hilfe einer UV LED oberhalb des Polymers gegeben, welcher zur Aushärtung führt. In y-Richtung wird mit Hilfe des Linienfokus (grün) zeitaufgelöst die Brechungsindexverteilung entlang des Linienfokus auf der Probe gemessen. Abbildung b) zeigt die Aufnahme vor der Belichtung. Entlang der Linie ist ein homogener Brechungsindex zu verzeichnen. Erfolgt nun der UV Puls, so kommt es zur Aushärtung und damit zu einer Brechungsindexänderung in diesem Bereich. Dies äußert sich im Bild durch die lokale Verschiebung des hell / dunkel Übergangs in x-Richtung (Abbildung c) Aufnahme nach ca. 0.5s). Der Brechungsindex steigt im belichteten Bereich weiter an und erreicht einen Maximalwert nach ca. 1 Sekunde (Abbildung d). Im weiteren folgt eine Verbreiterung des Aushärtebereichs in y-Richtung, bis letztendlich ca. 1/2 des betrachteten Bereiches in y-Richtung ausgehärtet ist. Auf diese Weise kann somit über die Brechungsindexverteilung die Kinetik der Aushärtung untersucht werden.

Nach einer entsprechenden Kalibrierung des Bildes, kann aus diesem die Information des lokalen schwarz / weiß Übergangs in eine Darstellung des Brechungsindexes gegenüber dem Ort aufgetragen wer-

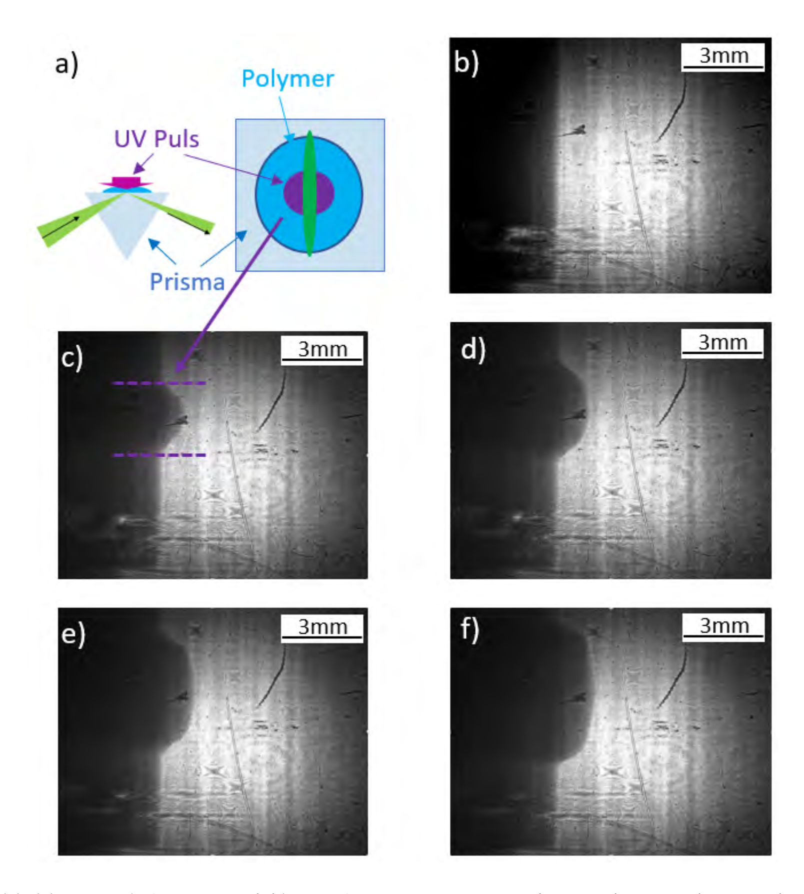


Abbildung 4: a) SFRIM und b) LineSRIM Vermessung der Brechungsindexverteilung nach der Aushärtung eines Polymers mit Hilfe eines Linienmusters einer UV DMD Beleuchtung.

den, wie es in Abbildung 5 dargestellt ist. Die Abbildung zeigt dabei den Brechungsindexverlauf entlang der y-Richtung vor der Belichtung (Start) und nach mehreren sequentiellen Belichtungen mit Hilfe des Projektors. Als Belichtungsmaske wurde dabei eine Sequenz von 10 angeschalteten und 5 ausgeschalteten Pixeln verwendet - beginnend mit 108 mJ/cm^2, gefolgt mit einer erhöhten Dosis von 215 mJ/cm^2 und abschließend eine Bestrahlung mit allen angeschalteten Pixeln bei 855 mJ/cm^2. Wie zu erkennen ist, zeigt sich noch eine klare Differenzierung zwischen aktiven und nicht aktiven Pixel und ein daraus resultierender ausgehärteter und nicht ausgehärteter Bereich für die geringste Energiedosis. Bei höherer Dosis zeigt sich ein Überstrahlen in eigentlich nicht beleuchtete Bereiche, so dass es auch dort zur Aushärtung kommt. Es zeigt sich aber auch, dass bei einer kompletten Beleuchtung der Probe, bei der alle Pixel aktiviert wurden, Variationen im Brechungsindex und damit im Aushärtegrad in den Bereichen der ursprünglich nicht aktiven Pixel vorliegen.

Zur Auswertung selbst wurden dabei 2 Vorgehensweisen gewählt – zum einen eine klassische Auswertung des Fresnel Fits und zum anderen die Auswertung über ein neuronales Netz, welches ein identisches Ergebnis liefert aber um 2 Größenordnungen schneller die Auswertung durchführt. Für den Aufbau des neuronalen Netzes wurde das „Neural Net Fitting“ Tool von Matlab genutzt. Die Größe der Eingangsschicht entspricht dabei der Pixelanzahl einer Bildzeile (1280). Im verwendeten Netz werden in den „hidden layers“ 50 Neuronen verknüpft. Die Ausgangsschicht reduziert das Ergebnis auf nur einen Ausgabewert, so dass jede Bildzeile mit einem Grauwertverlauf über 1280 Pixel eine Position der hell - dunkel Kante zugeordnet wird, was dem Brechungsindex entspricht. Die zum Anlernen des Netzes notwendigen Trainingsdaten wurden synthetisch über den theoretisch bekannten Verlauf der Fresnel-Reflexionen generiert.

4 Zusammenfassung

Zusammengefasst wurde ein bildbasiertes Messsystem für die örtlich und zeitlich aufgelöste Vermessung der Brechungsindexverteilung realisiert. Dieses kann für die Untersuchung des Aushärteverhaltens im Bereich der additiven Fertigung von optischen Komponenten einge-

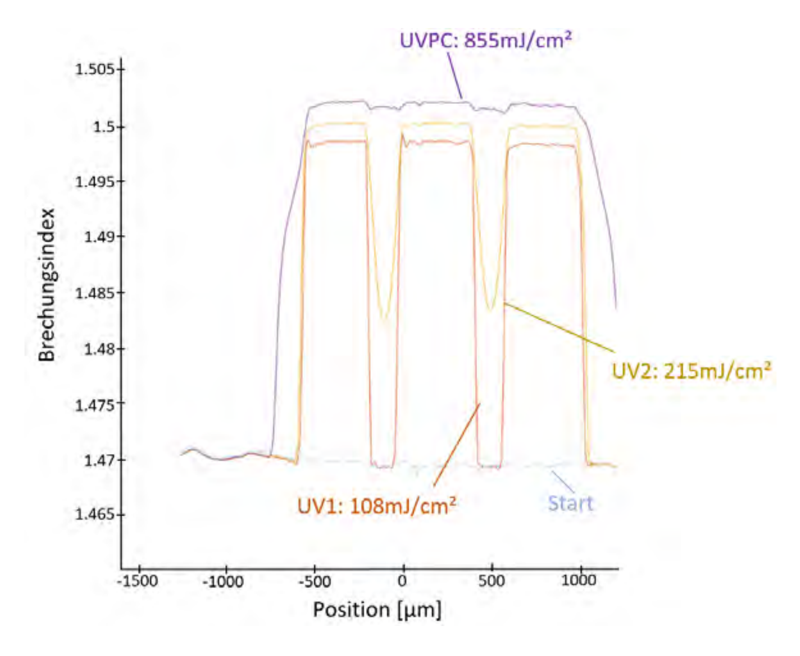


Abbildung 5: Zeitliche Entwicklung der Brechungsindex-Verteilung während der UV Aushärtung eines Polymers. (Start: blaue Linie; rote und gelbe Linie: on / off Modus einzelner Pixelreihen bei 108mJ/cm² und 215 mJ/cm²; violette Linie: alle Pixel des UV Projektes aktiviert)

setzt werden. Eine wesentliche Beschleunigung der Auswertung ergab sich dabei durch den Einsatz eines neuronalen Netzes, so dass das System auch eine „online“ Auswertung des Messergebnisses für zeitlich aufgelöste Vorgänge ermöglicht.

Literatur

1. M. Rank and A. Heinrich, “Measurement and use of the refractive index distribution in optical elements created by additive manufacturing,” in *Proc. 10930, Advanced Fabrication Technologies for Micro/Nano Optics and Photonics; doi: 10.1117/ 12.2507262*, Aalen, Germany, May 2019.
2. A. Heinrich, “3d printing of optical components,” in *3D Printing of Opti-*

cal Components, Springer Series in Optical Sciences, ISBN 978-3-030-58959-2, Aalen, Germany, March 2021.

3. T. Sun, Q. Ye, and X. Wang, "Scanning focused refractive-index microscopy," in *Sci Rep 4, 5647 https://doi.org/10.1038/srep05647*, Tianjin, China, July 2014.





Bildverarbeitung ist definitionsgemäß die Wissenschaft von der Verarbeitung von Bildern. Damit verknüpft das Fachgebiet die Sensorik von Kameras – bildgebender Sensorik – mit der Verarbeitung der aufgenommenen Sensordaten – den Bildern. Aus dieser Verknüpfung resultiert der besondere Reiz dieser Disziplin. Bildern begegnet der Mensch ständig, schon weil das Sehen die wichtigste Informationsquelle als Handlungsgrundlage für den Menschen bildet.
Der vorliegende Tagungsband des „Forums Bildverarbeitung“, das am 24. und 25. November 2022 in Karlsruhe als gemeinsame Veranstaltung des Instituts für Industrielle Informationstechnik am KIT und des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildauswertung stattfand, enthält die schriftlichen Aufsätze der eingegangenen Beiträge. Darin wird über aktuelle Trends und Lösungen der Bildverarbeitung in den methodischen Schwerpunkten Bildgewinnung, Qualitätssicherung, Sortierung, Bildverarbeitung, Fahrzeuge sowie Mess- und Automatisierungstechnik.

ISSN 2510-7224
ISBN 978-3-7315-1237-0



Gedruckt auf FSC-zertifiziertem Papier